Spectral Graph Drawing

Diplomarbeit
von
Thomas Puppe

Betreuer: Prof. Dr. Ulrik Brandes

Universität Konstanz
Fachbereich Mathematik und Statistik
Fachbereich Informatik und Informationswissenschaft

Januar 2005

Hiermit erkläre ich, daß ich die Diplomarbeit selbständig verfasst habe und keine anderen als die angegebenen Hilfsmittel und Quellen benutzt wurden.

Konstanz, den 07.01.2005

Thomas Puppe

Contents

Chapter 0

Zusammenfassung

In dieser Arbeit betrachten wir Zeichnungen von ungerichteten, gewichteten Graphen $G = (V, E, \omega)$. Zu jedem Graphen korrespondieren eine Reihe von Matrizen, die wir kurz vorstellen wollen:

- Die *Adjazenzmatrix* $A = (\omega_{ij})$, wobei ω_{ij} das Gewicht der Kante zwischen den Knoten i und j ist. Existiert keine Kante, so ist $\omega_{ij} = 0$.
- Die *Gradmatrix* $D = \text{diag}(d_i)$ ist eine Diagonalmatrix mit dem Knotengrad d_i am i-ten Diagonaleintrag. Der Knotengrad eines Knotens ist die Summe der Gewichte der dem Knoten adjazenten Kanten.
- Die *Laplacematrix* $L = D - A$.
- Die *relaxierte Laplacematrix* $L_\rho = (1 - \rho)D - A$, wobei ρ eine beliebige reelle Zahl ist.
- Die *generalisierte Laplacematrix* $L_G = D^{-1}A$, wobei wir voraussetzen, daß alle Knotengrade positiv sind.
- Die *normalisierte Laplacematrix* $L_N = D^{-\frac{1}{2}}L^{-\frac{1}{2}}$, wobei wir ebenfalls wieder voraussetzen, daß alle Knotengrade positiv sind.

Bis auf die generalisierte Laplacematrix L_G sind alle Matrizen reell symmetrisch und haben somit reelle Eigenwerte und -vektoren. Die Eigenwerte und -vektoren des generalisierten Eigenwertproblems $Lx = \lambda Dx$ entsprechen aber denen von L_G und sind reell, wenn alle Grade positiv sind.
Einerseits gilt, daß orthonormale Eigenvektoren einer Matrix B die quadratische Form

$x^T Bx$ minimieren (Satz von Rayleigh-Ritz). Andererseits gilt:

$$x^T Lx = \sum_{\substack{(i,j)\in E \\ i\neq j}} \omega_{ij}(x_i - x_j)^2 \quad .$$

Interpretiert man den Vektoreintrag x_i als Position des Knotens i in einer Zeichnung, dann entspricht die quadratische Form der gewichteten Summe der Kantenlängen. Hieraus folgt das zentrale Konzept spektralen Graphenzeichnens: Eigenvektoren gewisser Matrizen minimieren die Kantenlängen. Also definiert man: Das p-dimensionale *Laplace Layout* eines Graphen besteht aus den p nichttrivialen Eigenvektoren von L mit den kleinsten Eigenwerten. Gut strukturierte Graphen können damit gut visualisiert werden, siehe figure 5.3. Weniger gut strukturierte Graphen werden oft in der Form eines dichten Clusters mit wenigen, lose verbunden Knoten weit weg am Rand wiedergegeben. Um das zu vermeiden, nimmt man statt L im *relaxierten Laplace Layout* L_ρ, bzw. L_G im *generalisierten Laplace Layout* und berechnet deren Eigenvektoren. Bei diesen Ansätzen wird das ungewünschte Verhalten in der Optimierungsvorschrift bestraft. Höhergradige Knoten wandern nun ein wenig nach außen und ziehen die lose verbundenen Knoten nach innen, siehe figure 5.4 und 5.7. Das Layout ist besser. Es stellt sich heraus, daß für gewisse Werte des Relaxationsfaktors ρ das relaxierte und das generalisierte Layout sehr ähnlich sind.
Zur Berechnung der Eigenvektoren verwenden wir die *orthogonale Iteration*, eine mehrdimensionale Potenziteration. Es wird i.A. erwartet, daß der Algorithmus die Eigenvektoren in der Reihenfolge der Größe der Eigenwerte liefert. Wir zeigen jedoch, daß der Algorithmus eng verknüpft ist mit der QR-Iteration. Dort gibt es Sonderfälle, in denen die Reihenfolge vertauscht ist. Der Beweis der QR-Iteration dient dann auch als Beweis für die orthogonalen Iteration. Mittels der dafür eingeführten normalisierten Laplacematrix lassen sich mit dem Algorithmus auch die generalisierten Eigenvektoren von L_G berechnen.
Am Schluß führen wir eine neue Methode ein, mit der sich Graphen, die sich mit der Zeit ändern, mit spektralen Mitteln gezeichnet werden können. Für jeden Zeitschritt immer wieder die statischen spektralen Algorithmen anwenden, hat Nachteile. Um Aufwand zu sparen, sollten die Informationen der alten Zeichnung wiederverwendet werden. Die neue Zeichnung soll sich auch nicht zu sehr von der alten Zeichnung unterscheiden, um es dem Betrachter leichter zu machen, Strukturen wiederzuerkennen. Weil Eigenvektoren bis auf wenige Ausnahmen stetig sind, kann beides erreicht werden. Die alten Eigenvektoren werden als Startnäherung in die orthogonale Iteration eingesetzt. Die Konvergenz ist wesentlich schneller und die Zeichnungen sind stetig. Zu Testzwecken konnten wir sogar eine flüssige Animation mit spektralen Zeichnungen erstellen.

Chapter 1

Introduction

A graph is a mathematical structure consisting of two sets: a set of vertices and a set of links between these vertices, called edges. The task of graph drawing is finding an intuitive visualization of this abstract structure. There are a lot of daily life examples, where this has been done for decades. Think of genealogical trees or bus and subway maps. From these kind of diagrams everybody gains the intended information, without knowledge of the mathematics behind. But drawing such diagrams manually is a complex and costly work. For this reason automation algorithms were invented, to be able to use computers for visualizations, which work more efficiently. Today these methods are widely spread in computer science (compilers, databases, network design, ...), graphical data analysis (social sciences, chemistry, ...) and information visualization in general (flow charts, diagrams, ...).
In each of these fields we have different visualization tasks and therefore different solutions. A tree is for example a connected graph without cycles[1], i.e. there is from every node exact one path (with all nodes distinct) to any other node. But what, if we have excellent tree visualization methods, but the family tree we have to draw is not cycle-free[2]?

We will show in this thesis, that spectral layouts are a more flexible way to visualize a graph: The idea is to take only general readability criteria into account. Let edges act as forces between nodes. The more nodes are attracting or repelling each other, the closer or the more distant they should be positioned in the layout. With respect to this we optimize the location of the nodes using spectral methods. This way basic structural information of all kinds of graphs is provided. The force model is very adaptive to modifications in the graph. Continuous changes cause in almost any case continuous changes in the layout. Therefore spectral methods are a proper method for dynamic graph draw-

[1]For formal definitions of graph, cycle, path and tree see section 2.1.
[2]This is the case, if two ancestors were related by blood, before they got children.

ing, i.e. drawing a graph that changes over time. They are easy to implement and with acceleration methods even large-scale networks can be laid out.

In chapter 2 we define our notations and recall fundamental facts from graph theory and linear algebra. We take a closer look at spectral theory in chapter 3. Some not so widely known results are noted, on which parts of our work are based. We present theory on the generalized eigenvalue problem, the continuity of eigenvectors and we extend Gershgorin's eigenvalue bounds. Graph related matrices and some of their properties are introduced in chapter 4. We simplify e.g. their associated quadratic forms and compute eigenvalue bounds. These matrices are needed to generate spectral layouts. We characterize different spectral layouts in chapter 5, starting with the classical Laplace layout. To visualize graphs satisfactury, which have less symmetries, modifications of the Laplace layout are needed. We analyse two newer approaches, which improve the layout's quality. In chapter 6 we state an algorithm that unifies the algorithms for all three spectral layout approaches. We prove the correctness and discuss convergence anormalities. In chapter 7 we adapt the methods developed in chapter 5 and 6 to dynamic graph drawing and use an animated graph as test instance.

Chapter 2

Basic Notations and Facts

2.1 Graph Theory

A *directed graph* $G = (V, E)$ consists of a finite nonempty set V, $|V| = n$, and a relation $E \subseteq V \times V$. The elements v of V are called *vertices* or *nodes*, the elements $e = (u, v)$ of E are called *edges*. A graph is called *undirected*, if the pairs $(u, v) \in E$ are unordered, i.e. $(u, v) \equiv (v, u)$. Two nodes u, v are called *adjacent*, if there is an edge (u, v) in G.
The *neighbourhood* $N(v)$ of a node v is defined as $N(v) = \{u | \, (u, v) \text{ or } (v, u) \in E\}$. The elements of $N(v)$ are called *neighbours* of v.
A *path* from node v_1 to node v_k is a sequence $(v_1, v_2, ..., v_k)$ of nodes such that (v_i, v_{i+1}) is an edge for $1 \leq i \leq k-1$. A *cycle* is a path with $v_1 = v_k$ and all other nodes pairwise different. The *length* of a cycle is the number of successive edges in the path. An edge (v, v) is a cycle of length 1 and called *trivial cycle* or *self-loop*. A graph is called *connected*, if for any pair of nodes u, v, $u \neq v$, there is a path from u to v or from v to u. A graph is called *strongly connected*, if for any pair of nodes u, v, $u \neq v$, there is a path from u to v and from v to u. Every connected, undirected graph is strongly connected.

An undirected connected graph without cycles is called *tree*. A tree has often one distinct node called *root*. Nodes of a tree, that are only connected with one other node, are called *leave*. The node u is called child and node v is called u's correspondending *parent*, if u and v are adjacent nodes of a tree and the shortest path of u to the root is larger than the shortes path of v to the root.
$H = (V_H, E_H)$ is called *subgraph* of $G = (V, E)$, if $V_H \subseteq V$ and $E_H \subseteq E$.
A graph $G = (V, E, \omega)$ is called *bipartite*, if there are two node partitions V', V'' with $V' \cup V'' = V$, $V' \cap V'' = \emptyset$, such that all edges of G consist of an element of V' and an element of V''.

We label the n vertices of a graph G as v_i, $1 \leq i \leq n$. Sometimes we just notate node i. We define a weight function ω for G. Every edge (v_i, v_j) is associated with a real, nonzero weight ω_{ij}. If there is no edge (v_k, v_l) we set $\omega_{kl} := 0$. If $\omega_{i,j} \equiv 1$ for all edges (v_i, v_j) of G, then the graph and its edges are called *unweighted*, otherwise *weighted*. We denote a weighted graph G with $G = (V, E, \omega)$

This thesis is on layouts of undirected graphs only, so from now on we use "graph" synonym for "undirected graph", if not mentioned otherwise. A directed graph can be treated as an undirected graph by omitting the order of the edges. We assume, without loss of generality, that our graphs are connected. If a graph is not connected, we work on its connected subgraphs. We further request the edges to be unique, no parallel edges are allowed. We allow weighted edges and self-loops.

The *degree* d_i of a node v_i is here defined as

$$d_i := \sum_{v_j \in N(v_i)} \omega_{ij} \quad .$$

So adding a self-loop with weight ω_{ii} increases the degree of the node v_i by ω_{ii}. The maximum and minimum degree of a graph G is denoted by $\Delta(G)$ and $\delta(G)$, respectively. A graph with all degrees equal is called *regular*.

2.2 Linear Algebra Basics

In this section we recapitulate some important linear algebra definitions and facts. If not mentioned otherwise the theory was taken from [HoJo], where a brief summary of linear algebra with focus on - as the title of the book says - matrix analysis can be found. For proofs and additional theory refer to the classical literature, for example [GvL] or [Fis].

2.2.1 Basic Definitions

We are working on the field $\mathbb{K}$ and always $\mathbb{K} = \mathbb{C}$ or $\mathbb{K} = \mathbb{R}$. Let $A = (a_{ij}) \in \mathbb{R}^{n \times n}$ or $\mathbb{C}^{n \times n}$ be a matrix and $x \in \mathbb{R}^n$ or $\mathbb{C}^n$ be a vector. a_{ij} is the ij-th entry of A and x_i the i-th entry of x. A^T stands for the transpose of A, x^T for the transpose of x. A^* and x^* stand for the conjugate complex transpose, respectively. I is the unit matrix and $e^{(i)}$ is the unit vector with 1 at the i-th entry and all other entries equal 0. The vector $\mathbf{1}$ has all entries equal 1 and the vector $\mathbf{0}$ has all entries equal 0. We sometimes use the Kronecker delta δ_{ij}, defined by

$$\delta_{ij} = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{otherwise} \end{cases} \text{, for } 1 \leq i, j \leq n \ .$$

We have for vectors $x, y \in \mathbb{C}^n$ the (complex) euclidian scalar product

$$x^* y \ := \ \sum_{i=1}^{n} \overline{x_i} y_i$$

and the (complex) euclidian vector norm

$$|| \, x|| \ := \ \sqrt{x^* x} \ = \ \sqrt{\sum_{i=1}^{n} \overline{x_i} x_i} \quad .$$

Analogue we have in a real context for vectors $x, y \in \mathbb{R}^n$ the (real) euclidian scalar product

$$x^T y \ := \ \sum_{i=1}^{n} x_i y_i$$

and the (real) euclidian vector norm

$$|| \, x|| \ := \ \sqrt{x^T x} \ = \ \sqrt{\sum_{i=1}^{n} x_i^2} \quad .$$

The euclidian distance for vectors $x, y \in \mathbb{K}^n$ is defined by

$$\text{dist}(x, y) \ := \ || \, x - y|| \quad .$$

A set of vectors $x^{(1)}, x^{(2)}, ..., x^{(k)} \in \mathbb{K}^n$ is called *orthogonal* and denoted by $x^{(1)} \perp x^{(2)} \perp ... \perp x^{(k)}$, if

$$(x^{(i)})^T x^{(j)} \ = 0 \, , \quad \text{for } 1 \leq i < j \leq k \quad .$$

The vectors are called *orthonormal*, if additionally $|| \, x^{(i)}|| = 1$ for $1 \leq i \leq k$. Orthogonal vectors are linearly independent. A matrix $U \in \mathbb{C}^{n \times n}$ with orthonormal column vectors is called *unitary matrix*. U is nonsingular and $U^* = U^{-1}$. A matrix $U \in \mathbb{R}^{n \times n}$ with orthonormal column vectors is called *orthonormal matrix*. U is nonsingular and $U^T = U^{-1}$.
For a matrix $A \in \mathbb{K}^{n \times n}$ a set of vectors $x^{(1)}, x^{(2)}, ..., x^{(k)} \in \mathbb{K}^n$ is called *A-orthogonal*, if

$$(x^{(j)})^T A x^{(i)} \ = 0 \, , \quad \text{for } 1 \leq i < j \leq k \quad .$$

For matrices $A = (a_{ij}) \in \mathbb{K}^{n\times n}$ we use the *Frobenius norm*

$$||\, A||_F := \sqrt{\sum_{i=1}^{n}\sum_{j=1}^{n} |a_{ij}|^2} \quad .$$

The *determinant* $\det(A)$ of a matrix $A \in \mathbb{K}^{n\times n}$ is defined as usual.

2.2.2 Eigentheory

Definition 2.1 (Eigenvalue Problem)
If $A \in \mathbb{C}^{n\times n}$, $x \in \mathbb{C}^n$ and $\lambda \in \mathbb{C}$ we consider the equation:

$$Ax = \lambda x\,, \qquad x \neq 0 \quad .$$

If a scalar λ and a nonzero vector x happen to satisfy this equation, then λ is called an *eigenvalue* of A and x is called an *eigenvector* of A associated with λ. The pair (λ, x) is called *eigenpair*.
The set of all λ, that are eigenvalues of A is called the *spectrum* of A. The *spectral radius* of A is the nonnegative real number $r_\rho(A) := \max\{|\lambda| \ : \ \lambda \text{ is eigenvalue of } A\}$. The *eigenspace* E_λ is the set of all eigenvectors x associated with the eigenvalue λ. The *characteristic polynomial* p_A is defined by:

$$p_A(t) := \det(tI - A) \quad .$$

Remarks:

- The set of the n roots of $p_A(t)$ coincides with the spectrum of A. A matrix of order n has most n different eigenvectors. Eigenvalues must not be unique. Their multiplicity matches their multiplicity as zeros of $p_A(t)$.

- If $A, B \in \mathbb{C}^{n\times n}$ and B is invertible, then A and $B^{-1}AB$ have the same eigenvalues and x is an eigenvector of A iff $B^{-1}x$ is an eigenvector of $B^{-1}AB$. Eigenvalues are invariant under basis transformation.

- The spectral radius $r_\rho(A)$ is just the radius of the smallest disc centered at the origin in the complex plane that includes all the eigenvalues of A.

- We denote the eigenvalues of a matrix A with λ_i^A, $i = 1, ..., n$, but often we omit the A in λ_i^A. If the eigenvalues are real, they are numbered in non-decreasing order, i.e. $\lambda_1^A \leq ... \leq \lambda_n^A$.

Theorem 2.2 (Schur Decomposition)
Given $A \in \mathbb{C}^{n\times n}$ with eigenvalues $\lambda_1, ..., \lambda_n$ in any prescribed order, there is a unitary matrix $U \in \mathbb{C}^{n\times n}$ such that

$$U^*AU = T = (t_{ij})$$

is upper triangular, with diagonal entries $t_{ii} = \lambda_i$, $i = 1, .., n$. All eigenpairs of A are eigenpairs of T.
This matrix decomposition is called *Schur decomposition*.

Lemma 2.3
Let $\lambda \in \mathbb{C}$ be an eigenvalue of $A \in \mathbb{C}^{n\times n}$ with correspondending eigenvector $x \in \mathbb{C}^n$. Then yields for $c \in \mathbb{C}$, $k \in \mathbb{N}$:

a) $(c\lambda, x)$ is an eigenpair of cA.

b) $(c + \lambda, x)$ is an eigenpair of $cI + A$.

c) (λ^k, x) is an eigenpair of A^k.

2.2.3 Real Symmetric Matrices

Definition 2.4 (Real Symmetric Matrix)
A matrix $A \in \mathbb{R}^{n\times n}$ is called *(real) symmetric*, iff $A^T = A$.

The complex pendant: A matrix $B \in \mathbb{C}^{n\times n}$ is called *Hermitian*, iff $B^* = B$. A linear combination of real symmetric matrices is always real symmetric.

Theorem 2.5 (Spectral Theorem for Real Symmetric Matrices)
Let $A \in \mathbb{R}^{n\times n}$ be a symmetric matrix. There is Schur decomposition $U^TAU = D$ of A with U real orthonormal and D a real diagonal matrix such that the diagonal elements of D are the eigenvalues of A and the column vectors of U the correspondending eigenvectors.

Remarks:

- All eigenvectors correspondending to different eigenvalues of A are orthogonal.

- The multiplicity of the eigenvalues matches the dimension of the eigenspace of their eigenvectors. If $\dim(E_\lambda) > 1$, then $\dim(E_\lambda)$ eigenvectors of λ may be chosen to be orthogonal to themselves and to all eigenvectors corresponding to other eigenvalues.
- The column vectors of U form an orthonormal basis.
- There is a similar decomposition with D real and diagonal for Hermitian matrices, but U is there in general orthonormal and complex.

Definition 2.6 (Positive Definitness)
A real symmetric matrix A is said to be *positive definite*, if

$$x^T Ax > 0 \quad \text{for all nonzero } x \in \mathbb{R}^n \quad .$$

If the strict inequality $x^T Ax > 0$ is weakenend to $x^T Ax \geq 0$, then A is said to be *positive semidefinite*.

Definition 2.7 (Leading Principal Minors)
Given is $A \in \mathbb{R}^{n\times n}$. We denote by A_i the submatrix of A determined by deleting the last $n - i$ rows and columns of A. The *leading principal minors* of A are the real numbers $\det A_i$, $1 \leq i \leq n$.

Theorem 2.8
Given is $A \in \mathbb{R}^{n\times n}$. The following expressions are equivalent:

a) A is positive definite.

b) All eigenvalues of A are positive.

c) All leading principal minors of A are positive.

d) There exists a nonsingular matrix $C \in \mathbb{R}^{n\times n}$ with $A = CC^T$.

Additionally is A positive semidefinite, iff all eigenvalues of A are nonnegative.

Positive definite matrices are nonsingular. These matrices don't have 0 as an eigenvalue and therefore their kernel is empty.

Theorem 2.9 (Rayleigh-Ritz)
Given is a symmetric matrix $A \in \mathbb{R}^{n \times n}$ with eigenvalues $\lambda_1, ..., \lambda_n$ and correspondending orthonormal eigenvectors $x^{(1)}, ..., x^{(n)}$. Then holds for all $x \in \mathbb{R}^n$

$$\lambda_1 = \min_{x \neq 0} \frac{x^T A x}{x^T x}$$
$$\lambda_n = \max_{x \neq 0} \frac{x^T A x}{x^T x} \quad .$$

For the nonextremal eigenvalues holds for all $x \in \mathbb{R}^n$

$$\lambda_k = \min_{\substack{x \neq 0 \\ x \perp x^{(1)}, ..., x^{(k-1)}}} \frac{x^T A x}{x^T x}, \quad k = 2, 3, ..., n$$
$$\lambda_{n-k} = \max_{\substack{x \neq 0 \\ x \perp x^{(n)}, ..., x^{(n-k+1)}}} \frac{x^T A x}{x^T x}, \quad k = 1, 2, ..., n-1 \quad .$$

The ratio $\frac{x^T A x}{x^T x}$ is called *Rayleigh-Ritz ratio* or *coefficient*.

Chapter 3

Spectral Methods

In this section we state some not so well known result from spectral theory, which become important during the thesis.

3.1 The Generalized Eigenvalue Problem

Definition 3.1 (Generalized Eigenvalue Problem)
If $A, B \in \mathbb{C}^{n\times n}$, $x \in \mathbb{C}^n$ and $\lambda \in \mathbb{C}$ we consider the equation:

$$Ax = \lambda Bx\,, \qquad x \neq 0 \quad .$$

If a scalar λ and a nonzero vector x happen to satisfy this equation, then λ is called a *generalized eigenvalue* of (A, B) and x is called an *generalized eigenvector* of (A, B) associated with λ.

Remark: If B is invertible, the generalized eigenvalue problem is equivalent to the eigenvalue problem

$$B^{-1}Ax = \lambda x\,, \qquad x \neq 0 \quad .$$

From now on we concentrate on special case of the generalized eigenvalue problem with some nice properties. We suppose that $A, B \in \mathbb{R}^{n\times n}$ are symmetric and B additionally

positive definite. The generalized eigenvectors and eigenvalues of (A, B) are now in any case the eigenvectors and eigenvealues of $B^{-1}A$. For these matrices we have further:

Theorem 3.2
Given are symmetric matrices $A, B \in \mathbb{R}^{n\times n}$. If B is positive definite then (A, B) has n real generalized eigenvalues $\lambda_1, ..., \lambda_n$ with correspondending linearly independent eigenvectors $x^{(1)}, ..., x^{(n)}$. Moreover $x^{(i)}$ and $x^{(j)}$ are B-orthogonal if $\lambda_i \neq \lambda_j$. If $\lambda_i = \lambda_j$, then $x^{(i)}$ and $x^{(j)}$ may be chosen to be B-orthogonal.

Proof:
By the spectral theorem 2.5 the matrix B can be written as

$$B = UD^2U^T \quad ,$$

where $U \in \mathbb{R}^{n\times n}$ is orthogonal and $D^2 = DD \in \mathbb{R}^{n\times n}$ is diagonal. Since B is positive definite, also D is real, diagonal and has full rank. Therefore the following expressions are equivalent:

$$\begin{aligned} Ax &= \lambda Bx \\ Ax &= \lambda UD^2U^Tx \\ D^{-1}U^TAx &= \lambda DU^Tx \\ D^{-1}U^TAUD^{-1}DU^Tx &= \lambda DU^Tx \\ D^{-1}U^TAUD^{-1}y &= \lambda y \quad \text{with } y = DU^Tx \quad . \end{aligned}$$

The matrix $D^{-1}U^TAUD^{-1}$ is symmetric, since $\left(D^{-1}U^TAUD^{-1}\right)^T = D^{-1}U^TAUD^{-1}$. It has n real eigenvalues $\lambda_1, ..., \lambda_n$ and n correspondending real eigenvectors $y^{(1)}, ..., y^{(n)}$. If (λ, y) is an eigenpair of $D^{-1}U^TAUD^{-1}$, then (λ, x) is and eigenpair of $B^{-1}A$. The eigenvectors $y^{(i)}$ and $y^{(j)}$ are orthogonal if $\lambda_i \neq \lambda_j$. If $\lambda_i = \lambda_j$, then $y^{(i)}$ and $y^{(j)}$ may be chosen to be orthogonal. If $y^{(1)}, ..., y^{(n)}$ are orthogonal, they are linearly independent. Since U as an orthogonal matrix and D as a diagonal matrix have full rank, the eigenvectors $x^{(1)}, ..., x^{(n)}$ are also linearly independent. From the orthogonality of $y^{(1)}, ..., y^{(n)}$ follows

$$\begin{aligned} (y^{(i)})^Ty^{(j)} = 0 \quad &\Leftrightarrow \quad (DU^Tx^{(i)})^TDU^Tx^{(j)} = 0 \\ &\Leftrightarrow \quad (x^{(i)})^TBx^{(j)} = 0 \quad , \end{aligned}$$

with $i \neq j$. So the eigenvectors of $B^{-1}A$ are B-orthogonal. □

Theorem 3.3
Given are symmetric matrices $A, B \in \mathbb{R}^{n\times n}$. Additonally B is positive definite. Let $\lambda_1, ..., \lambda_n$ denote the generalized eigenvalues of (A, B) with correspondending B-orthogonal eigenvectors $x^{(1)}, ..., x^{(n)}$. Then holds for all $x \in \mathbb{R}^n$

$$\lambda_1 = \min_{x\neq 0} \frac{x^T Ax}{x^T Bx}$$
$$\lambda_n = \max_{x\neq 0} \frac{x^T Ax}{x^T Bx} \quad .$$

For the nonextremal eigenvalues holds for all $x \in \mathbb{R}^n$

$$\lambda_k = \min_{\substack{x\neq 0 \\ x^T Bx^{(1)}=0,...,x^T Bx^{(k-1)}=0}} \frac{x^T Ax}{x^T Bx}, \quad k = 2, 3, ..., n$$
$$\lambda_{n-k} = \max_{\substack{x\neq 0 \\ x^T Bx^{(n)}=0,...,x^T Bx^{(n-k+1)}=0}} \frac{x^T Ax}{x^T Bx}, \quad k = 1, 2, ..., n-1 \quad .$$

Proof:
Let the orthogonal matrix $U \in \mathbb{R}^{n\times n}$ and the diagonal matrix $D \in \mathbb{R}^{n\times n}$ be analogue defined as in the last proof. We have

$$\begin{aligned} Ax &= \lambda Bx \\ \Leftrightarrow \quad D^{-1}U^T AUD^{-1}y &= \lambda y \quad \text{with } y = DU^T x \quad . \end{aligned}$$

Now consider the Rayleigh-Ritz coeffizient of $D^{-1}U^T AUD^{-1}y$ with $y = DU^{-1}x$:

$$\begin{aligned} \frac{y^T D^{-1}U^T AUD^{-1}y}{y^T y} &= \frac{(UD^{-1}y)^T AUD^{-1}y}{y^T y} = \\ \frac{x^T Ax}{(DU^T x)^T DU^T x} &= \frac{x^T Ax}{x^T Bx} \end{aligned}$$

Now the assertion follows with theorem 2.9. □

3.2 Gershgorin's Discs and Extensions

This section is on a classical result, Gershgorin's disc theorem, and some extensions.

Definition 3.4
Let $A \in \mathbb{C}^{n\times n}$ be given. The *deleted absolute row sums* R_i of A are

$$R_i := \sum_{\substack{j=1 \\ j\neq i}}^{n} |a_{ij}| \quad , \quad 1 \leq i \leq n \quad .$$

Theorem 3.5 (Gershgorin)
All eigenvalues of the matrix $A \in \mathbb{C}^{n,n}$ are located in the union of the n discs

$$\bigcup_{i=1}^{n} \{z \in \mathbb{C} \,\big|\, |z - a_{ii}| \leq R_i\} := G(A) \quad .$$

(see [HoJo, th. 6.1.1])

Brauer's extension of Gershgorin's disc theorem becomes useful for our purposes later. It is not so widely known, so we state it together with a proof.

Theorem 3.6 (Brauer)
Let $A \in \mathbb{C}^{n\times n}$. All the eigenvalues of A are located in the $n(n-1)/2$ ovals of Cassini

$$\bigcup_{\substack{i,j=1 \\ i\neq j}}^{n} \{z \in \mathbb{C} \,\big|\, |z - a_{ii}|\,|z - a_{jj}| \leq R_i R_j\} := C(A) \quad .$$

Proof:
Let λ be an eigenvalue of $A = (a_{ij})$ with eigenvector $x \neq 0$. There is an element x_p of x that has largest absolute value. We may assume that $x_p = 1$. If all other entries of x are zero, then $\lambda = a_{pp}$. Since all diagonal elements of A are in (3.1), λ is in there, too.
If the other components of x are not all zero, let x_q be the component with the second largest absolute value, i.e.,

$$1 = |\,x_p| \geq |\,x_q| > 0 \quad \text{and} \quad |\,x_q| \geq |\,x_l| \quad \text{for all } l \neq r \text{ and } l \neq q \quad .$$

With the two indices p and q, the associated components from $Ax = \lambda x$ satisfy

$$(\lambda - a_{pp})x_p = \sum_{\substack{k=1 \\ k\neq p}}^{n} a_{pk}x_k \quad \text{and} \quad (\lambda - a_{qq})x_q = \sum_{\substack{k=1 \\ k\neq q}}^{n} a_{qk}x_k \quad .$$

We take absolute values above and have (cf. Definition 3.4)

$$|\,\lambda - a_{pp}| \leq \sum_{\substack{k=1 \\ k\neq p}}^{n} |\,a_{pk}| \cdot |\,x_k| \leq R_p \cdot |x_q|$$

$$|\,\lambda - a_{qq}| \cdot |\,x_q| \leq \sum_{\substack{k=1 \\ k\neq q}}^{n} |\,a_{qk}| \cdot |\,x_k| \leq R_q \cdot |\,x_p| = R_q \quad .$$

Multiplying these inequalities gives

$$|\,\lambda - a_{pp}| \cdot |\,\lambda - a_{qq}| \cdot |\,x_q| \leq R_p \cdot R_q \cdot |\,x_q| \quad ,$$

but as $|\,x_q| > 0$, then $|\,\lambda - a_{pp}| \cdot |\,\lambda - a_{qq}| \leq R_p \cdot R_q$. □

Lemma 3.7
Let $A \in \mathbb{C}^{n\times n}$. Then $C(A) \subseteq G(A)$, i.e. Brauer's eigenvalue approximation is equal to Gershgorin's or better.

Proof:
We have to show for an arbitrary $z \in \mathbb{C}$ and $1 \leq i < j \leq n$, that

$$(|z - a_{ii}|\,|z - a_{jj}| \leq R_iR_j) \Rightarrow (|z - a_{ii}| \leq R_i \text{ or } |z - a_{jj}| \leq R_j) \quad .$$

We assume that from the left side follows the contraposition of the right side, that is $|z - a_{ii}| > R_i$ and $|z - a_{jj}| > R_j$. Multiplicating both right side expressions results in $|z - a_{ii}|\,|z - a_{jj}| > R_iR_j$. This is a contradiction to the left side. □

Definition 3.8 (Indicator Matrix)
The *indicator matrix* $M(A) = (m_{ij}) \in \mathbb{R}^{n\times n}$ of a matrix $A = (a_{ij}) \in \mathbb{C}^{n\times n}$ is defined by

$$m_{ij} := \begin{cases} 1 & \text{if } a_{ij} \neq 0 \\ 0 & \text{if } a_{ij} = 0 \end{cases} \quad , 1 \leq i, j \leq n \quad .$$

There is an isomorphism between the indicator matrix $M(A) \in \mathbb{R}^{n\times n}$ and a (directed) unweighted graph $G = (V, E)$, $V = \{v_1, ..., v_n\}$, iff: $m_{ij} = 1 \Leftrightarrow (v_i, v_j) \in E$. Further matrix-graph relations will be covered in section 4.7.
Brauer's theorem depends on using two different rows of a given matrix at a time. An obvious generalization idea is using more than two rows at a time. But this does not work in general, for counterexamples refer to [HoJo, expr. 6.4.13-6.4.15]. But under certain conditions related to graph theory it is possible, as we will see now:

Theorem 3.9 (Brualdi)
Suppose all nodes of the underlying graph of $M(A)$ of a matrix $A \in \mathbb{C}^{n\times n}$ are part of a non-trivial cycle. Then every eigenvalue of A is contained in the region

$$\bigcup_{\gamma \text{ is a cycle in } G} \left\{ z \in \mathbb{C} \,\middle|\, \prod_{v_i \in \gamma} |z - a_{ii}| \leq \prod_{v_i \in \gamma} R_i \right\} \quad .$$

The notation means that if $\gamma = (v_{i_1}, v_{i_2}), ..., (v_{i_k}, v_{i_{k+1}})$ is a nontrivial cycle with $v_{i_{k+1}} \equiv v_{i_1}$, then each of the products contains exactly k terms, and the index i takes on the k values $i_1, ..., i_k$. (see [HoJo, th. 6.4.18])

Later on we will use Gershgorin's and Brauer's results for eigenvalue bounds. Brualdi's theorem needs much information about the structure of the matrix and its related graphs. The computation would be too expensive.

3.3 Perturbation Theory

In this section we study the influence of perturbations in a matrix on the spectrum and on the set of eigenvectors.

Theorem 3.10
Given is a matrix $A(t) = (a_{ij}(t)) \in \mathbb{C}^{n\times n}$, whose elements are continuous functions of a parameter $t \in \mathbb{C}$. Then the eigenvalues of $A(t)$ are continuous, too.

Proof:
The eigenvalues of $A(t)$ are the zeros of their characteristic polynomials $p_{A(t)}(\lambda) :=$

$\det(\lambda I - A(t))$. The characterisitic polynomials are continuous as a combination of continuous functions in the elements of A. Their zeros are therefore continuous, too. □

The continuity of the eigenvalues is also reflected in the following equations:

$$\begin{aligned} \det(A) &= \prod_{i=1}^{n} \lambda_i^A \\ \sum_{i=1}^{n} a_{ii} &= \sum_{i=1}^{n} \lambda_i^A \quad . \end{aligned}$$

A proof can be found in [HoJo, th. 1.2.12]. The next theorem shows, that the eigenvalue problem of Hermitian or real symmetric matrices is *perfectly conditioned*. That means that the perturbation in the eigenvalues is bounded by a term of the same order as the perturbation in the matrix. Therefore eigenvalue algorithms are numerical stable.

Theorem 3.11 (Hoffmann-Wielandt)
Let $A = (a_{ij})$ and $B = (b_{ij})$ be Hermitian or real symmetric matrices of order n. Let $\lambda_1^A, ..., \lambda_n^A$ be the eigenvalues of A and $\lambda_1^B, ..., \lambda_n^B$ be the eigenvalues of B. Then

$$\sum_{i=1}^{n} (\lambda_i^A - \lambda_i^B)^2 \leq \sum_{i=1}^{n} \sum_{j=1}^{n} |a_{ij} - b_{ij}|^2 = || A - B ||_F^2$$

(for a proof see [Fie, th. 9.21])

Weyl's theorem is another important estimate on eigenvalue perturbations. It follows from the Courant-Fisher theorem, a theorem similar to the Rayleigh-Ritz theorem (2.9). A proof and some extensions can be found in [HoJo, section 4.3].

Theorem 3.12 (Weyl)
Let A and B be Hermitian or real symmetric matrices of order n. Let λ_i^A, λ_i^B, and λ_i^{A+B} be arranged in increasing order. For each $k = 1, ..., n$ we have

$$\lambda_k^A + \lambda_1^B \leq \lambda_k^{A+B} \leq \lambda_k^A + \lambda_n^B$$

Parlett [Pa, pp. 14-15] shows, that for eigenvectors the situation is more delicate:

Theorem 3.13
Let $A, A' \in \mathbb{R}^{n\times n}$ be symmetric and $Ax = \lambda_0^A x$, $A'y = \mu y$ with $x, y \in \mathbb{R}^n$ and $\lambda_0^A, \mu \in \mathbb{R}$. The eigenvalue μ is separated from A's eigenvalues other than λ_0^A by a gap $\gamma := \min |\, \lambda_i^A - \mu|$, $1 \le i \le n$ and $\lambda_i^A \neq \lambda_0^A$. Then yields

$$\sin \angle(x, y) \le || \, A - A' \, || \, / \gamma \quad .$$

Let a symmetric matrix $A(t) \in \mathbb{R}^{n\times n}$ be given, whose elements are continuous functions of a parameter $t \in \mathbb{R}$. If for $t \in I$, I an interval, the eigenvalues of $A(t)$ retain their multiplicity, then there is a constant lower bound for γ and the eigenvectors are continuous. Without a gap eigenvectors can be very sensitive functions of the data. If for t_0 former distinct eigenvalues become a multiple eigenvalue (or a multiple eigenvalue becomes distinct), then there is no guarantee that the normalized eigenvectors vary continuously in a neighbourhood of t_0. Consider the following example constructed by Givens, where we have a discontinuity for $t = 0$:

$$A(t) := \begin{pmatrix} 1 + t\cos(2/t) & t\sin(2/t) \\ t\sin(2/t) & 1 - t\cos(2/t) \end{pmatrix}$$

Eigenvalues: $\{1 + t, 1 - t\}$

Eigenvectors: $\begin{pmatrix} \cos(1/t) \\ \sin(1/t) \end{pmatrix}, \begin{pmatrix} \sin(1/t) \\ -\cos(1/t) \end{pmatrix}$.

But such discontinuities are not necessary. In section 4.4 we give an example for matrix, that depends on a factor ρ and state two eigenvalues with eigenvectors. For a certain ρ, the eigenvalues become equal, but the eigenvectors remain continuous.
To measure the distance of a vector from being an eigenvector of a symmetric matrix we define the residuum:

Definition 3.14 (Residuum)
Given is $A \in \mathbb{R}^{n\times n}$ symmetric and $q \in \mathbb{R}^n$, $q \neq \mathbf{0}$. Then $r(A, q)$, the *residuum* of A and q, is defined by

$$r(A, q) := || \, Aq - \frac{q^T Aq}{q^T q} q || \quad .$$

The next theorem shows, that the Rayleigh-Ritz coefficient $\frac{q^T Aq}{q^T q}$ is the best choice for the "approximate eigenvalue" of q. A proof can be found in [Pa, p. 12].

Theorem 3.15
Given is $A \in \mathbb{R}^{n \times n}$ symmetric and $q \in \mathbb{R}^n$, $q \neq \mathbf{0}$. Then holds for all $c \in \mathbb{R}$:

$$|| Aq - \frac{q^T Aq}{q^T q} q|| \leq || Aq - cq|| \quad .$$

If q is an eigenvector of A, then the Rayleigh-Ritz coefficient is equal to the corresponding eigenvalue. Otherwise the residuum is an upper bound for the distance between the coefficient and A's closest eigenvalue (see [Pa, p. 69]):

Theorem 3.16
Given is a symmetric matrix $A \in \mathbb{R}^{n \times n}$ and a unit vector $q \in \mathbb{R}^n$. Let λ be the closest eigenvalue of A to $q^T Aq = \frac{q^T Aq}{q^T q}$, the Rayleigh-Ritz coefficient of q. Then yields:

$$| \lambda - q^T Aq| \quad \leq \quad r(A, q) = || Aq - (q^T Aq)q|| \quad .$$

If the eigenvalues lie not too dense, the residuum is also a good measure for the distance of a vector from being an eigenvector (see [Pa, pp. 222-223]):

Theorem 3.17
Given is a symmetric matrix $A \in \mathbb{R}^{n \times n}$ and a normal vector $q \in \mathbb{R}^n$. Let λ_0 be the closest eigenvalue of A to $q^T Aq$, the Rayleigh-Ritz coefficient of q. Let x be its corresponding eigenvector. The Rayleigh-Ritz coefficient $q^T Aq$ is separated from A's eigenvalues other than λ_0 by a gap $\gamma := \min | \lambda_i^A - q^T Aq|$, $1 \leq i \leq n$ and $\lambda_i^A \neq \lambda_0$. Then yields

$$| \sin \angle(x, q)| \quad \leq \quad r(A, q)/\gamma \quad .$$

Chapter 4

Graph Related Matrices

In this chapter we define some graph related matrices and present their basic properties. Commonly used in graph theory are only the adjacency matrix A and the Laplace matrix L. The degree matrix D is needed for the definition of all other matrices except for A. The relaxed Laplace matrix L_ρ was introduced in [BW] to visualize bibliographic networks. In [Ko] the generalized eigenvectors of (L, D) are used for graph drawing. Since the matrix $D^{-1}L =: L_G$ has the same vectors as (normal) eigenvectors, we call L_G the generalized Laplace matrix. For the computation of L_G we will need the normalized Laplace matrix L_N.

4.1 Adjacency Matrix

Definition 4.1 (Adjacency Matrix)
The *adjacency matrix* $A(G) = (a_{ij}) \in \mathbb{R}^{n\times n}$ of a graph $G = (V, E, \omega)$ is defined by

$$a_{ij} = \begin{cases} \omega_{ij} & \text{if there is an edge } (v_i, v_j) \\ 0 & \text{otherwise} \end{cases} .$$

We will often omit the G in $A(G)$.

An equivalent definition for the adjacency matrix A is: $A := (\omega_{ij})$. The adjacency matrix is sometimes defined only for unweighted graphs, e.g. in [GR], but most results carry over to the weighted definition. The indicator matrix (definition 3.8) is an unweighted adjacency matrix. The adjacency matrix is always real symmetric, since our graphs are undirected.

Theorem 4.2 (Perron-Frobenius)
Suppose A is a adjacency matrix of an undirected, connected graph G with nonnegative weights. Then:

a) The spectral radius $r_\rho(A)$ is a simple eigenvalue of A. If x is an eigenvector for $r_\rho(A)$, then no entries of x are zero, and all have the same sign.

b) Suppose $A_1 \in \mathbb{R}^{n\times n}$ has nonnegative components and $A - A_1$ has also nonnegative components. Then $r_\rho(A_1) \leq r_\rho(A)$, with equality iff $A_1 = A$.

c) If λ is an eigenvalue of A and $|\lambda| = r_\rho(A)$, then $\frac{\lambda}{r_\rho(A)}$ is an m-th root of unity and $e^{2\pi iq/m}\rho(A)$ is an eigenvalue of A for all q. Further, all cycles in G have length divisible by m.

The Perron-Frobenius theorem in this form is taken from [GR, th. 8.8.1], where also a) and b) are proven. A proof of c) can be found in e.g. [BP, th. 2.2.20, def. 2.2.26 and th. 2.2.30].

4.2 Degree Matrix

Theorem 4.3 (Degree Matrix)
The *degree matrix* $D(G) \in \mathbb{R}^{n\times n}$ of a graph $G = (V, E, \omega)$ is a diagonal matrix, i.e. all off-diagonal entries are zero, with degree d_i of the node v_i at the i, i-th entry. We will often omit the G in $D(G)$.

Remark: D is real symmetric. For $1 \leq i \leq n$ holds:

- If all degrees are nonzero, the degree matrix D is invertible. The inverse D^{-1} is a diagonal matrix with $\frac{1}{d_i}$ at the i, i-th entry.
- If all degrees are positive, we define D^y, $y \in \mathbb{R}$ as the diagonal matrix with $(d_i)^y$ at the i, i-th entry.
- D has as eigenvalues d_i and as corresponding eigenvectors the unit vectors e_i.
- D is by theorem 2.8 positive definite, iff $d_i > 0$ (for all i).

Lemma 4.4
Given is a graph $G = (V, E, \omega)$ and its degree matrix D. Then yields for all $x \in \mathbb{R}^n$:

$$x^T Dx = \sum_{\substack{(i,j)\in E \\ i\neq j}} \omega_{ij} \left(x_i^2 + x_j^2\right) + \sum_{i=1}^{n} \omega_{ii} x_i^2 \quad .$$

Proof:

$$\begin{aligned}
x^T Dx &= \sum_{i=1}^{n} d_i x_i^2 \\
&= \sum_{i=1}^{n}\sum_{j=1}^{n} \omega_{ij} x_i^2 \\
&= \sum_{i=1}^{n} \omega_{ii} x_i^2 + \sum_{i=2}^{n}\sum_{j=1}^{i-1} \omega_{ij} x_i^2 + \sum_{j=2}^{n}\sum_{i=1}^{j-1} \omega_{ij} x_i^2 \\
&= \sum_{i=1}^{n} \omega_{ii} x_i^2 + \sum_{i=2}^{n}\sum_{j=1}^{i-1} \omega_{ij} x_i^2 + \sum_{i=2}^{n}\sum_{j=1}^{i-1} \omega_{ji} x_j^2 \\
&= \sum_{i=1}^{n} \omega_{ii} x_i^2 + \sum_{i=2}^{n}\sum_{j=1}^{i-1} \omega_{ij} \left(x_i^2 + x_j^2\right) \\
&= \sum_{\substack{(i,j)\in E \\ i\neq j}} \omega_{ij} \left(x_i^2 + x_j^2\right) + \sum_{i=1}^{n} \omega_{ii} x_i^2
\end{aligned}$$

□

4.3 Laplace Matrix

Definition 4.5 (Laplace Matrix)
The *Laplace matrix* $L(G) \in \mathbb{R}^{n\times n}$ of a graph $G = (V, E, \omega)$ is defined as

$$L(G) := D(G) - A(G) = \begin{pmatrix} d_1 - \omega_{11} & -\omega_{12} & \cdots & -\omega_{1n} \\ -\omega_{21} & d_2 - \omega_{22} & \cdots & -\omega_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ -\omega_{n1} & -\omega_{n2} & \cdots & d_n - \omega_{nn} \end{pmatrix} .$$

We will often omit the G in $L(G)$.

As an linear combination of two real symmetric matrices L is real symmetric. The vector $\mathbf{1}$ is an eigenvector with 0 as eigenvalue, since the sum of all entries in each row of L is zero. We consider the diagonal elements of L:

$$d_i - \omega_{ii} = \left(\sum_{j\in N(i)} \omega_{ij} \right) - \omega_{ii} = \sum_{\substack{j\in N(i) \\ j\neq i}} \omega_{ij}$$

So self-loops have no influence on L.

Lemma 4.6
Given is a graph $G = (V, E, \omega)$ with Laplace matrix). Then yields for all $x \in \mathbb{R}^n$:

$$x^T Lx = \sum_{(i,j)\in E} \omega_{ij}(x_i - x_j)^2 \quad .$$

Proof:

$$
\begin{aligned}
x^T L x &= x^T D x - x^T A x \\
&= x^T D x - \sum_{i=1}^{n}\sum_{j=1}^{n} \omega_{ij} x_i x_j \\
&= \left(\sum_{i=1}^{n} \omega_{ii} x_i^2 + \sum_{i=2}^{n}\sum_{j=1}^{i-1} \omega_{ij} \left(x_i^2 + x_j^2 \right) \right) - \qquad \text{(lemma 4.4)} \\
&\quad \left(\sum_{i=1}^{n} \omega_{ii} x_i^2 + 2 \sum_{i=2}^{n}\sum_{j=1}^{i-1} \omega_{ij} x_i x_j \right) \\
&= \sum_{i=2}^{n}\sum_{j=1}^{i-1} \omega_{ij} (x_i^2 - 2 x_i x_j + x_j^2) \\
&= \sum_{(i,j)\in E} \omega_{ij} (x_i - x_j)^2
\end{aligned}
$$

□

A more elegant version of this proof restricted to nonnegative edge weights can be found in [Mo97, propos. 2.2]. From Lemma 4.6 follows that if all edge weights of a graph G are nonnegative, L is positive semidefinite.
For further theory on the Laplace matrix refer to [Mo91] and [Mo97].

4.4 Relaxed Laplace Matrix

Definition 4.7 (Relaxed Laplace Matrix)
The *relaxed Laplace matrix* $L(G) \in \mathbb{R}^{n\times n}$ of a graph $G = (V, E, \omega)$ is defined as

$$L_\rho(G) := (1-\rho)D(G) - A(G) = L(G) - \rho D(G) =$$

$$\begin{pmatrix} (1-\rho)d_1 - \omega_{11} & -\omega_{12} & \cdots & -\omega_{1n} \\ -\omega_{21} & (1-\rho)d_2 - \omega_{22} & \cdots & -\omega_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ -\omega_{n1} & -\omega_{n2} & \cdots & (1-\rho)d_n - \omega_{nn} \end{pmatrix}$$

for a *relaxation factor* $\rho \in \mathbb{R}$, mostly $\rho \in [0, 1]$. We will often omit the G in $L(G)$.

It is also possible to define L_ρ as Laplace matrix, whose diagonal entries are multiplicated with $(1-\rho)$. In this case self-loops would have no influence on L_ρ. This is equivalent to our definition, if we ignore self-loops in the graphs.
As a linear combination of two real symmetric matrices L_ρ is also real symmetric. Since $L_\rho = D - A - \rho D + \rho A - \rho A = (1-\rho)L - \rho A$, the matrix L_ρ compromises between the Laplace matrix L and the negative adjacency matrix $-A$. $L_0 = L$ and $L_1 = -A$. If G is connected and has nonnegative weights, then the smallest eigenvalue of L_ρ is simple (see [GR, lemma 13.9.1]). This yields for any ρ, in particular also for the Laplace matrix.

Lemma 4.8
Given is a relaxed Laplace matrix L_ρ of a graph $G = (V, E, \omega)$. Then for all $x \in \mathbb{R}^n$:

$$x^T L_\rho x = \sum_{\substack{(i,j)\in E \\ i\neq j}} \omega_{ij} \left((x_i - x_j)^2 - \rho x_i^2 - \rho x_j^2 \right) - \rho \sum_{i=1}^{n} \omega_{ii} x_i^2 \quad .$$

Proof:
Because of $L_\rho = L - \rho D$ this Lemma follows from Lemma 4.4 and Lemma 4.6. □

Lemma 4.8 shows that the relaxed Laplace matrix is in general not positive semidefinite, even if all weights of the underlying graph are nonnegative. The next two lemmas are on the properties of L_ρ for regular graphs G, i.e. if all degrees of G are equal.

Lemma 4.9
Given is a graph G with Laplace matrix L and the relaxed Laplace matrix L_ρ. Then following is equivalent:

a) The matrices L and L_ρ have the the same eigenvectors.

b) The vector $\mathbf{1}$ is an eigenvector of L_ρ.

c) The graph G is regular or $\rho = 0$.

Proof:
a) $\Rightarrow$ b) is clear. b) $\Rightarrow$ c) holds, since

$$L_\rho \mathbf{1} = \begin{pmatrix} \vdots \\ d_i - \rho d_i - \sum_{j=1}^{n} \omega_{ij} \\ \vdots \end{pmatrix} = -\rho \begin{pmatrix} \vdots \\ d_i \\ \vdots \end{pmatrix} \quad .$$

c) $\Rightarrow$ a) holds, because if $\rho = 0$, then $L = L_\rho$. And if G regular, then $L_\rho = L - \rho d I$, $d \in \mathbb{R}$ the node degree of G. So lemma 2.3 ensures, that L and L_ρ have the same eigenvectors. $\square$

Lemma 4.10
Suppose G is regular. Then L_{ρ_1} has λ as eigenvalue with eigenvector x, iff L_{ρ_2} has $\lambda + (\rho_2 - \rho_1)\Delta$ as eigenvalue with eigenvector x.

Proof:
The assertion follows with lemma 2.3 and

$$L_{\rho_1} = L - \rho_2 \Delta I + \rho_2 \Delta I - \rho_1 \Delta I = L_{\rho_2} + (\rho_2 - \rho_1)\Delta I \quad .$$

$\square$

Now we study the eigenvalues of L_ρ. If ρ is changed continuously, then the eigenvalues are changing continuously, too. So every eigenvalue λ of L_ρ can be expressed as a continuous function $f_\lambda(\rho)$, $\rho \in [0,1]$. The functions $f_\lambda(\rho)$ are strictly monotonic decreasing:

Lemma 4.11
Given are two matrices $L_{\rho_1}(G)$ and $L_{\rho_2}(G)$, $\rho_1 > \rho_2 \geq 0$, of the same underlying graph G with positve degrees. Then yields for $1 \leq k \leq n$:

$$\lambda_k^{L_{\rho_1}} < \lambda_k^{L_{\rho_2}} \quad .$$

Proof:
Set $A = (\rho_2 - \rho_1)D$ and $B = L_{\rho_2}$ in Weyl's theorem (3.12). □

It is possible, that there is an intersection between different eigenvalue functions. Consider the matrix L_ρ of the graph in figure 4.1. For simplicity we set $q := (1-\rho)$:

$$L_\rho = \begin{pmatrix} 2q & -1 & -1 & & & & \\ -1 & 2q & -1 & & & 0 & \\ -1 & -1 & 3q & -1 & & & \\ & & -1 & 2q & -1 & & \\ & & & -1 & 3q & -1 & -1 \\ & 0 & & & -1 & q & 0 \\ & & & & -1 & 0 & q \end{pmatrix} \quad .$$

One eigenvector of L_ρ is $x = (0,0,0,0,0,1,-1)^T$ for every ρ with eigenvalue function $f_1(\rho) = q = 1-\rho$. Another eigenvalue has the function $f_2(\rho) = (1-2\sqrt{2})\rho + 3 - \sqrt{2}$ with corresponding eigenvector $y(\rho) = (y_1, ..., y_7)^T$, defined by

$$\begin{aligned} y_7 &:= 1 \\ y_6 &:= y_7 \\ y_5 &:= q - f_2 \\ y_4 &:= (3q - f_2)y_5 - 2y_6 \\ y_3 &:= (2q - f_2)y_4 - y_5 \\ y_2 &:= \frac{1}{2}\left((3q - f_2)y_3 - y_4\right) \\ y_1 &:= y_2 \end{aligned}$$

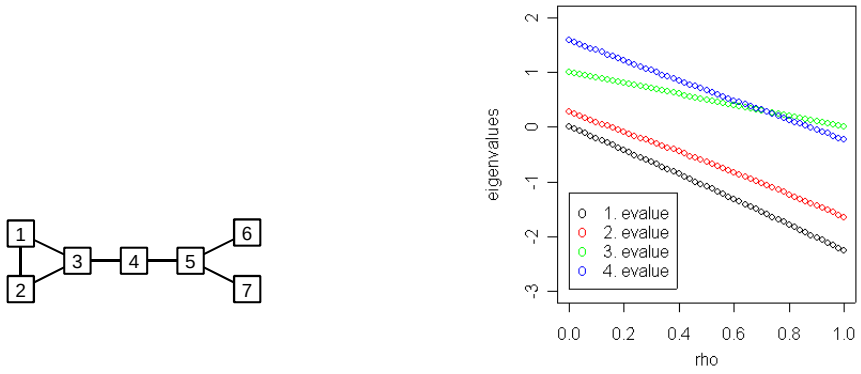


Figure 4.1: Eigenvalue intersection. On the left the graph, on the right the eigenvalues of its matrix L_ρ.

The vector $y(\rho)$ is an solution of the linear system $(L_\rho - f_2 I)y = \mathbf{0}$ and therefore are $(y(\rho), f_2(\rho))$ an eigenpair of L_ρ for all ρ. The eigenvalue functions $f_1 =$ and f_2 intersect for $\rho = \frac{1}{2}\sqrt{2}$. To the eigenvalue $1 - \frac{1}{2}\sqrt{2}$ correspond the orthogonal eigenvectors x and $y = (-2 + 2\sqrt{2}, -2 + 2\sqrt{2}, -2 + \sqrt{2}, -2, 0, 1, 1)^T$. The function f_1 is the green line in the eigenvalue diagram in figure 4.1, and f_2 the blue line.
As long as a graph is connected and its weights are nonnegative, the smallest eigenvalue is simple and there will be no intersection of the smallest eigenvalue function with any other.

4.5 Generalized Laplace Matrix

Definition 4.12 (Generalized Laplace Matrix)
The *generalized Laplace matrix* $L_G(G) \in \mathbb{R}^{n \times n}$ of a graph $G = (V, E, \omega)$ with positive degrees is defined as

$$L_G(G) := D(G)^{-1} L(G) =$$

$$\begin{pmatrix} \frac{d_1 - \omega_{11}}{d_1} & \frac{-\omega_{12}}{d_1} & \cdots & \frac{-\omega_{1n}}{d_1} \\ \frac{-\omega_{21}}{d_2} & \frac{d_2 - \omega_{22}}{d_2} & \cdots & \frac{-\omega_{2n}}{d_2} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{-\omega_{n1}}{d_n} & \frac{-\omega_{n2}}{d_n} & \cdots & \frac{d_n - \omega_{nn}}{d_n} \end{pmatrix} .$$

We will often omit the G in $L_G(G)$.

Analogue to L_ρ self-loops could be ignored in this definition of L_G. The generalized Laplace matrix L_G of an undirected graph G is symmetric, iff G is regular, i.e. all nodes have degree $d \neq 0$. Then L_G is equal $\frac{1}{d}L$ and has the same eigenvectors as L. The vector $\mathbf{1}$ is in any case an eigenvector of L_G with eigenvalue 0. Since all degrees of G are positive, the inverse degree matrix D^{-1} is positive definite. The eigenvalues and eigenvectors of L_G are the generalized eigenvalues of (L, D). All eigenvalues of L_G are real with correspondending real and D-orthogonal eigenvectors (see theorem 3.2).

Theorem 4.13
Given is a generalized Laplace matrix L_G with positive degrees. Then yields for all vectors $x \in \mathbb{R}^n$, $x \neq 0$:

$$x^T L_G x = \sum_{(i,j) \in E} \omega_{ij} \left(\frac{x_i}{d_i} - \frac{x_j}{d_j} \right) (x_i - x_j) \quad .$$

Proof:

$$\begin{aligned}
x^T L_G x &= \sum_{i=1}^{n} \frac{1}{d_i} \left(d_i x_i^2 - \sum_{j=1}^{n} \omega_{ij} x_i x_j \right) \\
&= \sum_{i=1}^{n} \frac{1}{d_i} \left(\sum_{j=1}^{n} \omega_{ij} x_i^2 - \sum_{j=1}^{n} \omega_{ij} x_i x_j \right) \\
&= \sum_{i=1}^{n} \sum_{j=1}^{n} \frac{\omega_{ij}}{d_i} \left(x_i^2 - x_i x_j \right) \\
&= \sum_{i>j} \frac{\omega_{ij}}{d_i} \left(x_i^2 - x_i x_j \right) + \sum_{j>i} \frac{\omega_{ij}}{d_i} \left(x_i^2 - x_i x_j \right) \\
&= \sum_{i>j} \frac{\omega_{ij}}{d_i} \left(x_i^2 - x_i x_j \right) + \sum_{i>j} \frac{\omega_{ji}}{d_j} \left(x_j^2 - x_j x_i \right) \\
&= \sum_{i>j} \omega_{ij} \left[\frac{x_i^2}{d_i} - \left(\frac{1}{d_i} + \frac{1}{d_j} \right) x_i x_j + \frac{x_j^2}{d_j} \right] \\
&= \sum_{(i,j) \in E} \omega_{ij} \left(\frac{x_i}{d_i} - \frac{x_j}{d_j} \right) (x_i - x_j)
\end{aligned}$$

□

Lemma 4.14
Given is a graph G with positive degrees, its generalized Laplace matrix L_G and its relaxed Laplace matrix L_ρ. Then yields for all $\rho \in \mathbb{R}$: $u \in \mathbb{R}^n$ is an eigenvector of L_G with eigenvalue λ, iff u is an eigenvector of $D^{-1}L_p$ with eigenvalue $\lambda - \rho$.

Proof:
The following expressions are equivalent:

$$\begin{aligned}
D^{-1}Lu &= \lambda u \\
Lu &= \lambda Du \\
Lu - \rho Du &= \lambda Du - \rho Du \\
L_\rho u &= (\lambda - \rho) Du \\
D^{-1} L_\rho u &= (\lambda - \rho) u
\end{aligned}$$

□

If all weights are nonnegative, the matrix $D^{-1}A$ is a so-called *stochastic* matrix. It has the same eigenvectors as L_G, but in a reversed order. Later on we will introduce graph layouts, for which eigenvectors of graph related matrices form the coordinates. Because an relaxation factor ρ does not change the set of eigenvectors of $D^{-1}L_\rho$, we will only consider $D^{-1}L = L_G$.

4.6 Normalized Laplace Matrix

Definition 4.15 (Normalized Laplace Matrix)
The *normalized Laplace matrix* $L_N(G) \in \mathbb{R}^{n\times n}$ of a graph $G = (V, E, \omega)$ with positive degrees is defined as

$$L_N(G) := D(G)^{-\frac{1}{2}} L(G) D(G)^{-\frac{1}{2}} =$$

$$\begin{pmatrix} \frac{d_1-\omega_{11}}{d_1} & \frac{-\omega_{12}}{\sqrt{d_2}\sqrt{d_1}} & \cdots & \frac{-\omega_{1n}}{\sqrt{d_n}\sqrt{d_1}} \\ \frac{-\omega_{21}}{\sqrt{d_1}\sqrt{d_2}} & \frac{d_2-\omega_{22}}{d_2} & \cdots & \frac{-\omega_{2n}}{\sqrt{d_n}\sqrt{d_2}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{-\omega_{n1}}{\sqrt{d_1}\sqrt{d_n}} & \frac{-\omega_{n2}}{\sqrt{d_2}\sqrt{d_n}} & \cdots & \frac{d_n-\omega_{nn}}{d_n} \end{pmatrix} .$$

We will often omit the G in $L_N(G)$.

Again the definition of L_N could be modified to ignore self-loops. Obviously is L_N symmetric and has therefore a real spectrum. 0 is an eigenvalue of L_N with eigenvector $D^{\frac{1}{2}}\mathbf{1}$. The next lemmas show the basic property of the normalized Laplace matrix, its spectral relation to the generalized Laplace matrix:

Lemma 4.16
Given is a graph $G = (V, E, \omega)$ with positve degrees and corresponding generalized Laplace matrix L_G and normalized Laplace matrix L_N. Then yields:

a) $L_G = D^{-\frac{1}{2}} L_N D^{\frac{1}{2}}$

b) (λ, u) is an eigenpair of L_N, iff $(\lambda, D^{-\frac{1}{2}} u)$ is an eigenpair of L_G.

Proof:
a)

$$\begin{aligned} L_G &= D^{-1} L \\ &= D^{-1} \left(D^{\frac{1}{2}} L_N D^{\frac{1}{2}} \right) \\ &= D^{-\frac{1}{2}} L_N D^{\frac{1}{2}} \end{aligned}$$

b) The following expressions are equivalent:

$$\begin{aligned} L_N u &= \lambda u \\ D^{-\frac{1}{2}} L_N D^{\frac{1}{2}} D^{-\frac{1}{2}} u &= \lambda D^{-\frac{1}{2}} u \\ L_G D^{\frac{1}{2}} u &= \lambda D^{-\frac{1}{2}} u \end{aligned}$$

□

With the same reasons as for L_G we do not introduce an relaxation factor for L_N. If G is connected and has nonnegative weights, then the smallest eigenvalue of L_N is simple (see [GR, lemma 13.9.1]). From the next lemma can be concluded that in this case L_N is positive semidefinite. With lemma 4.16 both carries over to L_G.

Lemma 4.17
Given is a normalized Laplace matrix L_N. For all $x \in \mathbb{R}^n$ yields

$$x^T L_N x = \sum_{(i,j) \in E} \omega_{ij} \left(\frac{x_i}{\sqrt{d_i}} - \frac{x_j}{\sqrt{d_j}} \right)^2 .$$

Proof:

$$x^T L_N x = \left(D^{-\frac{1}{2}} x \right)^T L \left(D^{-\frac{1}{2}} x \right) .$$

□

4.7 Isomorphisms

Definition 4.18 (Permutation Matrix)
A *permutation matrix* $P \in \mathbb{R}^{n \times n}$ is a matrix with one component equal to 1 and all other zero in each row and column.

Permutation matrices P are nonsingular and orthogonal. Left-multiplication of matrix A by P permutes the rows, right-multiplication permutes the columns. The term PAP^T permutes rows and columns the same way.

Definition 4.19
Two graphs $G = (V, E, \omega)$ and $G' = (V', E', \omega')$ are *isomorphic*, if there is a bijective node permutation $\phi : V \mapsto V'$ and for the edges holds $(u, v) \in E \Leftrightarrow (\phi(u), \phi(v)) \in E'$ with $\omega_{uv} = \omega'_{\phi(u)\phi(v)}$.

Theorem 4.20
Given is a graph $G = (V, E, \omega)$, a relaxation factor ρ and the correspondending adjacency matrix $A(G)$, degree matrix $D(G)$, Laplace matrix $L(G)$, relaxed Laplace matrix $L_\rho(G)$, generalized laplace matrix $L_G(G)$ and normalized Laplace matrix $L_N(G)$.

a) There is by definition an isomorphism between G and the matrices $A(G)$, $L(G)$, $L_G(G)$ and $L_N(G)$. There is also an isomorphism between (G, ρ) and $L_\rho(G)$.

b) Given is a graph $G' = (V', E', \omega')$ isomorphic to G. Then $A(G')$, $D(G')$, $L(G')$, $L_\rho(G')$, $L_G(G')$ and $L_N(G')$ have the same spectrum as $A(G)$, $D(G)$, $L(G)$, $L_\rho(G)$, $L_G(G)$ and $L_N(G)$, respectively. The associated eigenvectors have permutated entries.

c) A class of isomorphic graphs is completely determined by the eigenvalues and correspondending orthogonal eigenvectors of the following matrices: $A(G)$, $L(G)$, $L_\rho(G)$ and $L_N(G)$. A class of isomorphic graphs is completely determined by the matrix D and the eigenvalues and correspondending D-orthogonal eigenvectors of L_G.

Proof:

a) Holds by definition.

Figure 4.2: Two cospectral graphs [GR, p. 164]

b) There is for each of the mentioned matrices $M(G)$ a permutation matrix P_M with $P_M M(G) P_M^T = M(G')$. If x is an eigenvector of $M(G)$ with eigenvector λ, then Px is an eigenvector of $M(G')$ with eigenvector λ (see remarks to definition 2.1).

c) Since the matrices $A(G)$, $L(G)$, $L_\rho(G)$ and $L_N(G)$ are real symmetric, we can build their Schur decomposition (theorem 2.5) with the given eigenvalues and eigenvectors. The eigenvalues and eigenvectors of L_N can be computed with the matrix D, the eigenvalues and eigenvectors of L_G and lemma 4.16.

□

The spectrum alone does not determine the underlying graph, even in the unweighted case. Figure 4.2 shows two different graphs, whose unweighted adjacency matrix has the same spectrum. Those graphs are called *cospectral*.

4.8 Eigenvalue Bounds

From now on we only work with matrices whose spectrum is real. We number their eigenvalues in non-decreasing order, i.e. $\lambda_1 \leq ... \leq \lambda_n$.
Later on we use improved upper bounds for the spectrum of graph related matrices to speedup the convergence of an algorithm. So we concentrate here on bounds in terms of the degree of nodes, because they are easy to compute. There are a lot of other eigenvalue bounds especially for graph related matrices. Chung [Ch, lem. 1.9 and 1.14] uses for example bounds for the generalized Laplacian matrix in terms of the diameter of a graph. The diameter is the maximum length of a shortest path between two nodes. Mohar uses the diameter [Mo91, eq 6.10.] for the Laplace matrix, as he also does with

another graph property: Weights of edges, which must be cut for a certain partitioning of the graph, see [Mo97, section 3.1]. In [GR, lem. 13.7.1 and cor. 13.7.2] other bounds for the Laplace matrix are stated also in terms of edge cuts. For Brualdi's extension of Gershgorin's lemma (theorem 3.9) the cycle structure of the graph must be given. Unfortunatly it is in practice often very costly to achieve knowledge of graph properties like diameter, minimal cuts or cycles.

4.8.1 Bounds of the Relaxed Laplace Matrix

First we treat eigenvalue bounds of the relaxed Laplace matrix L_ρ. The negative adjacency matrix $-A$ and the Laplace matrix L are special cases of L_ρ and therefore their eigenvalue bounds, too. A negative upper bound of $-A$ is a lower bound of A and a negative lower bound of $-A$ is an upper bound of A. We denote the maximum degree of the underlying graph as Δ, the minimum degree as δ and the deleted row sums R_i as in definition 3.4 :

$$R_i := \sum_{\substack{j=1 \\ j \neq i}}^{n} |\omega_{ij}| \quad , \quad 1 \leq i \leq n \quad .$$

The matrix L_ρ has real spectrum. So Gershgorin's theorem (3.5) gives us the following theorem:

Theorem 4.21
Given is a relaxed Laplace matrix L_ρ of a graph $G = (V, E, \omega)$. Then

$$\min_{1 \leq i \leq n} (1-\rho)d_i - \omega_{ii} - R_i \quad \leq \quad \lambda^{L_\rho} \quad \leq \quad \max_{1 \leq i \leq n} (1-\rho)d_i - \omega_{ii} + R_i$$

for all eigenvalues λ^{L_ρ} of L_ρ. Suppose now, that all weights are nonnegative and the underlying graph has no self-loops. If $\rho \in [0, 2]$, the bounds can be written as

$$-\rho\Delta \ \leq \ \lambda^{L_\rho} \ \leq \ (2-\rho)\Delta \quad .$$

Under the same assumptions holds for the eigenvalues λ^L of the Laplace matrix

$$0 \ \leq \ \lambda^L \ \leq \ 2\Delta$$

and for the eigenvalues λ^A of the adjacency matrix

$$-\Delta \ \leq \ \lambda^A \ \leq \ \Delta \quad .$$

If $\rho < 0$, then

$$-\rho\delta \;\leq\; \lambda^{L_\rho} \;\leq\; (2-\rho)\Delta \quad ,$$

and if $\rho > 2$, then

$$-\rho\Delta \;\leq\; \lambda^{L_\rho} \;\leq\; (2-\rho)\delta \quad .$$

These bounds can be sharpened using the extensions of Gershgorin's theorem:

Theorem 4.22
Given is a relaxed Laplace matrix L_ρ with all weights nonnegative, $\rho \in [0,1]$ and without self-loops. Let Δ_2 denote the second-largest degree of L_ρ. If there are two nodes in the underlying graph with degree Δ, then $\Delta_2 := \Delta$. Then holds for all eigenvalues λ^{L_ρ} of L_ρ:

$$-\rho\Delta \;\leq\; \frac{1}{2}\left((1-\rho)(\Delta+\Delta_2) - \sqrt{(1-\rho)^2(\Delta-\Delta_2)^2 + 4\Delta\Delta_2}\right) \;\leq\; \lambda^{L_\rho} \;\leq$$
$$\frac{1}{2}\left((1-\rho)(\Delta+\Delta_2) + \sqrt{(1-\rho)^2(\Delta-\Delta_2)^2 + 4\Delta\Delta_2}\right) \;\leq\; (2-\rho)\Delta \quad .$$

Under the same assumptions holds for the eigenvalues λ^L of the Laplace matrix

$$0 \;\leq\; \lambda^L \;\leq\; \Delta + \Delta_2$$

and for the eigenvalues λ^A of the adjacency matrix

$$-\sqrt{\Delta\Delta_2} \;\leq\; \lambda^A \;\leq\; \sqrt{\Delta\Delta_2} \quad .$$

Proof:
Theorem 3.1 yields that all eigenvalues of L_ρ are contained in

$$\bigcup_{\substack{i,j=1 \\ i\neq j}}^{n} K_{ij} \quad \text{with}$$
$$K_{ij} := \{x \in \mathbb{R} : |\, x-(1-\rho)d_i| \cdot |\, x-(1-\rho)d_j| \leq d_i d_j \,\} \quad .$$

We want to find the minium and the maximum of the above expression. We define for $d_i \geq d_j$

$$\begin{aligned} a &:= (1-\rho)d_i \\ b &:= (1-\rho)d_j \\ c &:= d_i d_j \end{aligned}$$

$$\begin{aligned} f_{ij}(x) &:= |x-a| \cdot |x-b| - c \\ &= \begin{cases} (x-a)(x-b)-c & \text{for } x \leq b \\ -(x-a)(x-b)-c & \text{for } b < x < a \\ (x-a)(x-b)-c & \text{for } a \leq x \end{cases} . \end{aligned}$$

Since all $d_i \geq 0$ and $\rho \in [0,1]$, we have

$$\begin{aligned} & a \geq b \geq 0 \\ & c = d_i d_j \geq (1-\rho)^2 d_i d_j = ab \geq 0 \quad . \end{aligned}$$

If $f_{ij} \leq 0$, then $x \in K_{ij}$. We compute the zeros of f_{ij} :

$$\begin{aligned} (x-a)(x-b)-c &= 0 \\ \Leftrightarrow \quad x_{1/2} &= \frac{1}{2}\left(a+b \pm \sqrt{(a+b)^2+4c-4ab}\right) \qquad (*) \\ \Leftrightarrow \quad x_{1/2} &= \frac{1}{2}\left(a+b \pm \sqrt{(a-b)^2+4c}\right) \end{aligned}$$

Since

$$x_1 = \frac{1}{2}\left(a+b+\sqrt{(a-b)^2+4c}\right) \geq \frac{1}{2}\left(a+b+\sqrt{(a-b)^2}\right) = a$$

and

$$x_2 = \frac{1}{2}\left(a+b-\sqrt{(a-b)^2+4c}\right) \leq \frac{1}{2}\left(a+b-\sqrt{(a-b)^2}\right) = b$$

x_1 and x_2 are two zeros of f_{ij}. It can easily be verified that there are no zeros for $b < x < a$ and that thus x_1 and x_2 are the only zeros. Because of

$$\lim_{x \to \pm\infty} f_{ij}(x) = +\infty \quad ,$$

$\max_{1 \leq i,j \leq n} x_1$ and $\min_{1 \leq i,j \leq n} x_2$ are an upper and a lower bound for all K_{ij} . To determine the maximum and the minimum of x_1 and x_2 consider again $(*)$:

$$\begin{aligned} x_{1/2} &= \frac{1}{2}\left(a+b \pm \sqrt{(a+b)^2+4c-4ab}\right) \\ &= \frac{1}{2}\left(a+b \pm \sqrt{(a+b)^2+4kab-4ab}\right) \quad , \text{where } kab = c\,,\ k \geq 1 \\ &= \frac{1}{2}\left(a+b \pm \sqrt{(a+b)^2+(k-1)4ab}\right) \quad . \end{aligned}$$

As $(k-1)4ab \geq 0$ we get the maximum and minimum of x_1 and x_2 if a and b become maximal, i.e. $a = \Delta$ and $b = \Delta_2$. For $\rho = 0$ and $\rho = 1$ we get the results for the Laplace matrix and the negative adjacency matrix, respectively. As shown in lemma 3.7 these bounds are always as least as good as Gershgorin's bounds. □

The bounds of the last theorem become as better as larger the difference between Δ and Δ_2 becomes.
The question is now, how good the upper and lower bounds we statet above approximate the spectrum. We will see, that certain graphs have these bounds as smallest and largest eigenvalue. Thus our bounds are in this sense tight. But first we prove an upper bound for the smallest eigenvalue and a lower bound for the largest one.

Theorem 4.23
Given is a relaxed Laplace matrix L_ρ with $\rho \in [0,1]$. Then

$$\begin{aligned} \lambda_1 &\leq \max_{1\leq i\leq n} -\rho d_i \\ \lambda_n &\geq \max_{1\leq i\leq n} (1-\rho)d_i - \omega_{ii} \quad . \end{aligned}$$

If the underlying graph has nonnegative degrees and no self-loops, then

$$\begin{aligned} \lambda_1 &\leq -\rho\delta \\ \lambda_n &\geq (1-\rho)\Delta \quad . \end{aligned}$$

Proof:
We get the lower bounds by setting $A = L$, $B = -\rho D$ and $k = 1$ in Weyl's theorem (3.12). With the Rayleigh-Ritz theorem (2.9) we have

$$\lambda_n \geq \frac{x^T L_\rho x}{x^T x} \quad \text{for all } x \neq 0\,.$$

The upper bounds follow by $x = e_i$, $i = 1, .., n$. □

Theorem 4.24
Given is a bipartite regular graph G with nonnegative edge weights. Then λ is eigenvalue of L_ρ, iff $2(1-\rho)\Delta - \lambda$ is eigenvalue of L_ρ . In particular

$$\lambda_1^{L_\rho} = -\rho\Delta \quad \text{and} \quad \lambda_n^{L_\rho} = (2-\rho)\Delta \quad .$$

Proof:
Bipartite graphs have no self-loops. Thus there exists for a bipartite graph G an isomorphic graph G' with relaxed Laplace matrix L_ρ of the form

$$\begin{pmatrix} \ddots & 0 & B \\ 0 & l_i & 0 \\ B^T & 0 & \ddots \end{pmatrix}, \; l_i = (1-\rho)d_i, \; B \in \mathbb{R}^{k \times m}, \; k+m=n \quad .$$

We partition the vector $x \in \mathbb{R}^n$ such that $x = (y,z)^T$, $y \in \mathbb{R}^k$ and $z \in \mathbb{R}^m$. Suppose x is an eigenvector of L_ρ with eigenvalue λ, i.e.

$$L_\rho \begin{pmatrix} y \\ z \end{pmatrix} = \lambda \begin{pmatrix} y \\ z \end{pmatrix} \quad .$$

Then

$$L_\rho \begin{pmatrix} y \\ -z \end{pmatrix} \;=\; \begin{pmatrix} \vdots \\ 2(1-\rho)d_i y_s - \lambda y_s \\ \vdots \\ \ldots\ldots\ldots\ldots\ldots\ldots \\ \vdots \\ 2(1-\rho)d_i(-z_t) - \lambda(-z_t) \\ \vdots \end{pmatrix} .$$

If $d_i = \Delta$ for $1 \leq i \leq n$, this can be simplified to

$$L_\rho \begin{pmatrix} y \\ -z \end{pmatrix} \;=\; (2(1-\rho)\Delta - \lambda) \begin{pmatrix} y \\ -z \end{pmatrix} \quad .$$

Since 0 is the smallest eigenvalue of the Laplace matrix L, the largest now must be 2Δ. The exact values of the extremal eigenvalues of L_ρ follow with lemma 4.10. □

The last theorem can be sharpened for the adjacency matrix using the Perron-Frobenius theorem (4.2). We state a theorem taken from [GvL, p. 178]. On the same page a proof is sketched for unweighted adjacency matrices only. But the proof carries over to the weighted case, if the weights are nonnegative.

Theorem 4.25
Let A be the adjacency matrix of a graph G and r_ρ its spectral radius. Then the following is equivalent:

a) G is bipartite.

b) The spectrum of A is symmetric around the origin, i.e. if λ is an eigenvalue, then also $-\lambda$ is an eigenvalue with the same multiplicity.

c) $-r_\rho$ is an eigenvalue.

The next theorem is an upper bound for the largest eigenvalue of a Laplace matrix with nonnegative weights and without self-loops. The bound is better than all other bounds stated here, but we found no generalization for L_ρ. Theorem and proof is taken from [AM].

Theorem 4.26
Given is a graph $G = (V, E, \omega)$ with nonnegative weights, without self-loops and the Laplace matrix $L(G)$. Then:

$$\lambda_n^L \leq \max\{d_i + d_j \,|\, (i,j) \in E\} \quad .$$

Proof (sketched):
Given is a graph $G = (V, E, \omega)$, $|V| = n$, $|E| = m$. We first introduce the vertex-edge *incidence matrix* $E(G) = (e_{ik}) \in \mathbb{R}^{n \times m}$ of G:

$$e_{ik} = \begin{cases} -\sqrt{\omega_{ij}} & \text{if there is an edge } k = (i,j) \\ \sqrt{\omega_{ij}} & \text{if there is an edge } k = (j,i) \\ 0 & \text{otherwise} \end{cases} \quad , i < j \, .$$

So in every row k of E there are the entries $-\sqrt{\omega_{ij}}$ and $\sqrt{\omega_{ij}}$ with $k = (i,j)$ and all other are zero. Now it can be shown that $EE^T = L$. We define another matrix $N := E^T E$, $N \in \mathbb{R}^{m \times m}$. If $Lx = \lambda x$ with $\lambda \neq 0$, then $NE^T x = E^T Lx = \lambda E^T x$, so that λ is also an eigenvalue of N. The sum of the absolute values of the row k of N equals $d_i + d_j$, if $k = (i,j)$. The assertion is the upper gershgorin bound of N. □

We found for L_ρ, $\rho \neq 0$, no matrix with similar properties as the incidence matrix has for L.

An important class of graphs later on are those with nonnegative weights and no self-loops. We can summarize their eigenvalue bounds of L_ρ, $\rho \in [0,1]$, with

$$\begin{aligned} -\rho\Delta \;&\leq\; \tfrac{1}{2}\left((1-\rho)(\Delta+\Delta_2) - \sqrt{(1-\rho)^2(\Delta-\Delta_2)^2 + 4\Delta\Delta_2}\right) \\ &\leq\; \lambda_1^{L_\rho} \;\leq \\ -\rho\delta \;&\leq\; (1-\rho)\Delta \\ &\leq\; \lambda_n^{L_\rho} \;\leq \\ \tfrac{1}{2}\left((1-\rho)(\Delta+\Delta_2) + \sqrt{(1-\rho)^2(\Delta-\Delta_2)^2 + 4\Delta\Delta_2}\right) \;&\leq\; (2-\rho)\Delta \quad . \end{aligned}$$

4.8.2 Bounds of the Generalized Laplace Matrix

Now we bound the eigenvalues of the generalized Laplace matrix L_G and the normalized Laplace matrix L_N. They have the same eigenvalues, as we have seen in lemma 4.16. We concentrate here on matrices, whose underlying graph has positive weights. Thus the spectrum of L_G and L_N is real and nonnegative. We state the bounds for L_G, but of course they also yield for L_N. If graphs have some negative edge weights, then Gershgorin's bounds should be choosen.

Theorem 4.27
Given is a graph G and its normalized Laplace matrix L_G. Then

$$\lambda_1^{L_G} = 0 \quad \text{and} \quad \max_{1\leq i\leq n} 1 - \frac{\omega_{ii}}{d_i} \;\leq\; \lambda_n^{L_G} \;\leq\; 2 \quad .$$

Proof:
We already know that 0 is the smallest eigenvalue of L_G. With theorem (3.3) we have

$$\lambda_n^{L_G} \geq \frac{x^T L x}{x^T D x} \quad \text{for all } x \neq 0\,.$$

We get the lower bound of $\lambda_n^{L_G}$ for $x = e_i$, $i = 1, ..., n$ in the above ratio. For the upper bound we consider

$$\frac{x^T L_N x}{x^T x} = \frac{y^T L y}{y^T D y} \quad \text{with} \quad y = D^{-\frac{1}{2}} x \quad .$$

We use lemma 4.6 and lemma 4.4 and get

$$\begin{aligned} \frac{y^T L y}{y^T D y} &= \frac{\sum_{(i,j)\in E} \omega_{ij}(y_i - y_j)^2}{\sum_{(i,j)\in E, i\neq j} \omega_{ij}\left(y_i^2 + y_j^2\right) + \sum_{i=1}^n \omega_{ii} y_i^2} \\ &\leq \frac{2\sum_{(i,j)\in E, i\neq j} \omega_{ij}(y_i^2 + y_j^2)}{\sum_{(i,j)\in E, i\neq j} \omega_{ij}\left(y_i^2 + y_j^2\right) + \sum_{i=1}^n \omega_{ii} y_i^2} \\ &\leq 2 \quad , \end{aligned}$$

since

$$(y_i - y_j)^2 \leq 2(y_i^2 + y_j^2) \quad .$$

$(i, j) \in E$ means here $\omega_{ij} \neq 0$, $i \leq j$. □

We now show that the bounds for L_G are tight with similar arguments as for L_ρ:

Theorem 4.28
Given is a bipartite graph G. Then λ^{L_G} is eigenvalue of L_G, iff $2 - \lambda^{L_G}$ is eigenvalue of L_G. In particular

$$\lambda_1^{L_G} = 0 \quad \text{and} \quad \lambda_n^{L_G} = 2 \quad .$$

Proof:
Bipartite graphs have no self-loops. Thus there exists for a bipartite graph G an isomorphic graph G' with generalized Laplace matrix L_G of the form

$$\begin{pmatrix} \ddots & 0 & B \\ 0 & 1 & 0 \\ B^T & 0 & \ddots \end{pmatrix}, B \in \mathbb{R}^{k\times m},\ k + m = n \quad .$$

We partition the vector $x \in \mathbb{R}^n$ such that $x = (y, z)^T$, $y \in \mathbb{R}^k$ and $z \in \mathbb{R}^m$. Suppose x is an eigenvector of L_G with eigenvalue λ^{L_G}, i.e.

$$L_G \begin{pmatrix} y \\ z \end{pmatrix} = \lambda^{L_G} \begin{pmatrix} y \\ z \end{pmatrix} \quad .$$

Then

$$L_G \begin{pmatrix} y \\ -z \end{pmatrix} \quad = \quad (2 - \lambda) \begin{pmatrix} y \\ -z \end{pmatrix} \quad .$$

Since 0 is the smallest eigenvalue of the Laplace matrix L_G, the largest must be 2. □

Chapter 5

Characterizing Spectral Graph Layouts

5.1 Motivation

We give a formal definition of graph layouts first:

Definition 5.1
Given is a graph $G = (V, E)$, $|V| = n$. A p-dimensional *layout* or *drawing* $l(G)$ of G is defined by n vectors $l^{(1)}, ..., l^{(n)} \in \mathbb{R}^p$, where $l^{(i)}$ is the coordinate vector of node v_i and $p \leq n$. We often omit the G in $l(G)$.
A layout $l(G)$ is completly determined also by its p *axis vectors* $x^{(i)} = \left(l_i^{(1)}, ..., l_i^{(n)}\right)^T$, $i = 1, ..., p$.

In general is $p \leq n$, but for the most visualization tasks is $p \leq 3$. Our convention for drawing edges is representing them simply by straight lines between their nodes. We do not need arrows, since our graphs are undirected. All underlying graphs of the layouts shown in this thesis will be unweighted.
To illustrate their relation we state now coordinate and axis vectors of the left graph layout in figure 5.1:

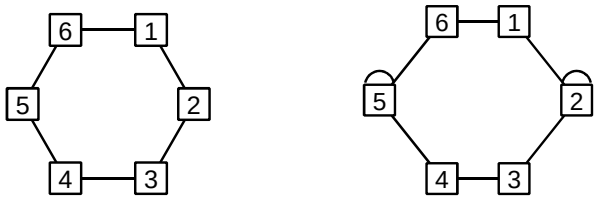


Figure 5.1: Two graphs in relaxed Laplace layouts.

coordinate vectors:
$l^{(1)} = (0.29, 0.5)^T$
$l^{(2)} = (0.58, 0)^T$
$l^{(3)} = (0.29, -0.5)^T$
$l^{(4)} = (-0.29, -0.5)^T$
$l^{(5)} = (-0.58, 0)^T$
$l^{(6)} = (-0.29, 0.5)^T$

axis vectors:
$x = (0.29, 0.58, 0.29, -0.29, -0.58, -0.29)^T$
$y = (0.5, 0, -0.5, -0.5, 0, 0.5)^T$

There are many contexts, in which graph drawing problems occur. Often we have for certain classes of graph certain layout goals. Typical requirements for tree drawings are for example:

> "Given is a tree T with root r. Find a two-dimensional layout for T without edge crossings, with integer vertex coordinates, with straight-line elements as edges and childs placed below their parents in the y-coordinate, ... "

For further tree drawing problems and solutions refer to [KW, chapter 3]. Sometimes are neither special design goals nor graph properties provided. Despite of this the layout of the graph should convey structural information in an intuitive way. According to [KW, p.71/72] we should take then the following general readability criteria into account:

1) Vertices should spread well on the page.

2) Adjacent vertices should be close.

Uniform vertex distribution reduces clutter. Close adjacent vertices imply moderate differences in edge lengths and leave an undistorted impression of the graph. One solution to generate such layouts is to model the graph by attracting or repelling forces between nodes. A configuration of such a system with minimal internal stress should optimize the quality of a layout. This approach is called *force-directed*. Some of this models are based on physical anologies, e.g. springs or magnetic fields. For more information refer to [KW, ch. 4] or [BETT, ch. 10].
In this thesis we consider positive edge weights as the amount of attracting forces and negative edge weights as the amount of repelling forces. Minimizing the internal stress[1] of this system means minimizing the length of positive weighted edges and maximizing the length of negative weighted edges. To measure the stress we define the quadratic weighted node distances as energy function for a graph layout. With lemma 4.6 we have:

Definition 5.2
Given is a graph $G = (V, E, \omega)$. The energy $E_L(p)$ of a p-dimensional layout l with axis vectors $x^{(i)}$, $1 \leq i \leq p$, is defined as

$$E_L(p) := \sum_{(i,j)\in E} \omega_{ij} || \, l^{(i)} - l^{(j)} ||^2 = \sum_{i=1}^{p} (x^{(i)})^T L x^{(i)} \quad .$$

If $E_L(p)$ is minimized, then the stress should also be minimized and hence readability criterion 2 should be guaranteed. In order to find local minima of E_L, we set the partial derivates of $x^T Lx$ for a single dimension equal zero:

$$\frac{\partial x^T L x}{\partial x_i} \;=\; 2 \sum_{j\in N(i)} \omega_{ij}(x_i - x_j) \;\stackrel{!}{=}\; 0 \quad .$$

This is equivalent to x_i, the coordinate of node i, being the *weighted centroid* of its neighbours:

$$x_i \;=\; \frac{\sum_{j\in N(i)} \omega_{ij} x_j}{d_i - \omega_{ii}} \;=:\; c_\omega(i) \quad .$$

[1] Edge weights could also be interpreted as the amount of node similarity or dissimilarity. The stress is minimized, if similar nodes are placed close together and dissimilar nodes are distant from each other.

When allowing only node i to move, then $c_\omega(i)$ is the optimal place with respect to E_L. If x is linear dependent on $\mathbf{1}$, then all node coordinates x_i are equal and therefore in $c_\omega(i)$. The layout would collapse in one single point contradictory to readability criterion 1. A method to avoid this is fixing the coordinates of a subset of nodes. Under this condition we can find layouts, in which every vertex not fixed in advance is placed in the barycenter of its neighbours. Such layouts are called *barycentric layouts* [KW, p. 79].
For another method to obtain a non-trivial drawing, we regard L's Rayleigh-Ritz coefficient $\frac{x^T L x}{x^T x}$ with $x \neq 0$. The coefficient becomes minimal, iff E_L becomes minimal. Let $x^{(i)}$ denote an eigenvector of L corresponding to the i-th smallest eigenvalue, $1 \leq i \leq n$. As showed in section 4.3 is $x^{(1)} = \mathbf{1}$ with eigenvalue 0. From theorem 2.9 we know, that

$$\lambda_k \quad = \quad \min_{\substack{x \neq 0 \\ x \perp x^{(1)}, \ldots, x \perp x^{(k-1)}}} \frac{x^T L x}{x^T x}, \quad k = 2, 3, ..., n \quad .$$

Eigenvectors corresponding to the smallest non-zero eigenvalues of L minimize the energy function E_L and are non-trivial. Thus they are a good choice for being axis vectors of a graph layout. Additionally their orthogonality ensures that the difference in the axis vectors is maximized and as much information as possible is provided with each new dimension. If they are chosen to be normal, the nodes are forced to be equally scattered in the drawing. So both readability criteria should be guaranteed. Graph layouts, whose axis vectors are orthonormal eigenvectors of graph related matrices, are called *spectral layouts*.
Another nice property of spectral layouts is, that an exact solution can be computed efficiently, whereas other approaches are often NP-hard problems, that only can be approximated. More about implementation is stated in chapter 6.
We now characterize layouts using eigenvectors of the Laplace matrix L. Then we continue with two newer approaches using eigenvectors of the generalized Laplace matrix L_ρ and the generalized eigenvalues of L, respectively. These eigenvectors minimize modified layout energy functions, but are still non-trivial, normal and to a certain degree orthogonal.

5.2 Laplace Layout

Let $G = (V, E, \omega)$ be a graph with Laplace matrix L. We give now a formal definition of the in the last section implicitly derived Laplace layout and characterize it as optimization problem. We will sometimes use the Kronecker delta δ_{ij}.

Definition 5.3
Assume that the eigenvalues of L are $\lambda_1^L \leq ... \leq \lambda_n^L$ with correspondending orthonormal eigenvectors $y^{(1)}, ..., y^{(n)} \in \mathbb{R}^n$. The p-dimensional *Laplace layout* of G is defined by p axis vectors $x^{(i)}$, where

$$x^{(i)} := y^{(i+1)} \quad , 1 \leq i \leq p \leq n \quad .$$

Theorem 5.4
A p-dimensional Laplace layout of a graph is the solution of the following constraint minimization problem:

$$\min_{x^{(1)},...,x^{(p)}} E_L(p)$$

$$\text{subject to: } (x^{(i)})^T x^{(j)} = \delta_{ij}\,, \quad 1 \leq i < j \leq p$$
$$\mathbf{1}^T x^{(i)} = 0\,, \quad 1 \leq i \leq p \quad .$$

Koren [Ko, pp. 9-10] reformulates this optimization problem as follows. Let the *variance* Var, the *standard deviation* σ and the *covariance* Cov be defined as usual:

$$\begin{aligned} \mathrm{Var}(y) &= \frac{1}{n}\sum_{i=1}^{n}(y_i - \overline{y})^2 \\ \sigma(y) &= \sqrt{\mathrm{Var}(y)} \\ \mathrm{Cov}(y^{(i)}, y^{(j)}) &= \frac{1}{n}\sum_{i=1}^{n}(y_i^{(i)} - \overline{y}^{(i)})(y_i^{(j)} - \overline{y}^{(j)})\,, \text{ where } \overline{y} \text{ is the mean of } y\,. \end{aligned}$$

Then yields:

Theorem 5.5
A p-dimensional Laplace layout l of a graph is the solution of the following constraint minimization problem:

$$\min_{x^{(1)},...,x^{(p)}} \frac{E_L(p)}{\sum_{i<j} ||\, l^{(i)} - l^{(j)}||^2} \;=$$

$$\min_{x^{(1)},...,x^{(p)}} \frac{\sum_{(i,j)\in E} \omega_{ij} ||\, l^{(i)} - l^{(j)}||^2}{\sum_{i<j} ||\, l^{(i)} - l^{(j)}||^2} \qquad (*)$$

$$\text{subject to: } \mathrm{Var}(x^{(1)}) = ... = \mathrm{Var}(x^{(p)}) \qquad (**)$$
$$\mathrm{Cov}(x^{(i)}, x^{(j)}) = 0\,, \quad 1 \leq i < j \leq p. \qquad (***)$$

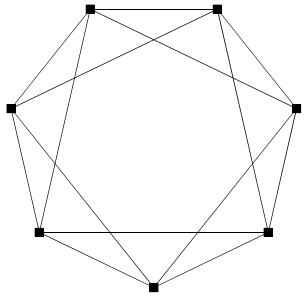


Figure 5.2: (7,1,2)-torus in a Laplace layout.

With this version the layout properties can nicely be characterized. If $(*)$ is minimized, the numerator is minimized and the denominator maximized. This means, that on the one hand the weighted edge lengths should become short and on the other hand the nodes should spread well in the drawing area in such a layout. So the general readability criteria from section 5.1 should be fulfilled. The first constraint $(**)$ ensures, that in each dimension the nodes are equally scattered. The second constraint $(***)$ ensures, that there is no correlation between the dimensions, i.e. that each dimension provides with as much information as possible. See figures 5.2 and 5.3 as examples.
For more generality we allow in this and the other spectral layouts negative weights as far as possible. But negative weights let edge lengths grow and may disturb the layout. So in practice using them extensively should be avoided. At least the degrees should remain positive.
We now define a class of very structured graphs:

Definition 5.6 (Torus)
Let $d, k \in \mathbb{N}$ and $V_k^d = \underbrace{V_k \times \cdots \times V_k}_{d \text{ times}}$. The graph $G = (V_k^d, E)$ with

$$\{(v_1, \ldots, v_d), (w_1, \ldots, w_d)\} \in E \quad \Leftrightarrow \quad 1 \leq \sum_{i=1}^{d} \min\{v_i - w_i, w_i - v_i \mod k\} \leq r$$

is called (k, d, r)-*torus*.

The graph in figure 5.2 is for example a (7,1,2)-torus. A (k,2,1)-torus would be a rectangular grid with k nodes, whose side nodes are connected with the opposite side nodes.

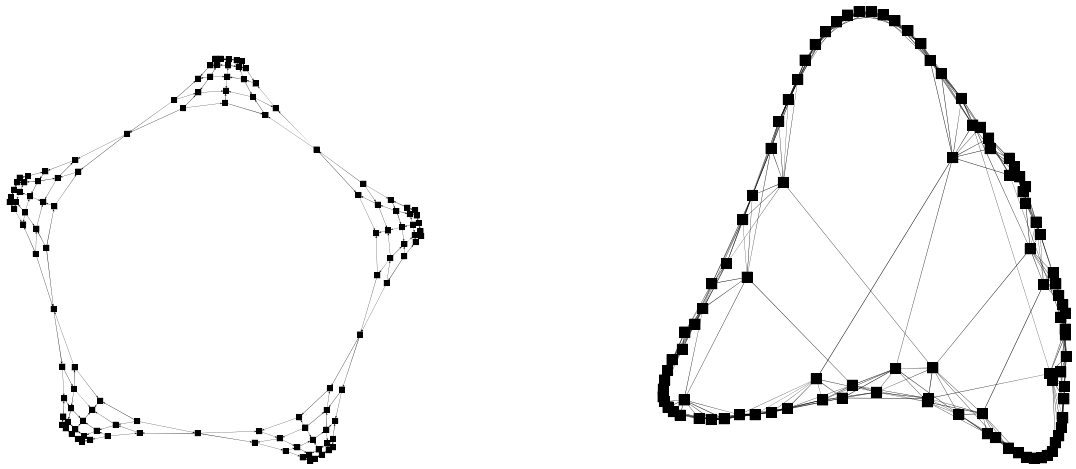


Figure 5.3: Two torus-like graphs in a Laplace layout. On the left a modified grid. On the right a small world ($n = 99$, $r = 4$, $p = 0.04$), see appendix A.

The Laplace layout can nicely display highly structured graphs and their symmetries. Their balanced structure forces the nodes to be equally scattered in the layout, if edge lengths become short. Examples are all kinds of torus-like graphs, see figure 5.3. The left graph was a rectangular grid in original. Some interior nodes and some at the boundary were removed. The graph has 120 nodes and 200 edges. The minimum degree is 2, the maximum 4, the mean is about 3.33 and the standard deviation about 0.62. The graph on the right was in original a (99,1,4)-torus. It has 99 nodes and 396 edges. Then with probability 0.04 each of this edges was disconnected from one of its nodes and reconnected with a random node. Its minimum degree is 7, the maximum 10, the mean is 8 and the standard deviation is about 0.49. Although the graph has low deviations in the degree and is only locally dense, the maximum over all shortest paths between two nodes, the diameter, is relative low. The graph is a so-called *small world* model (see appendix A). In the graph here the force-model of the Laplace layout is very apparent. Without the reconnected edges the layout would look like a circle, as in figure 5.2. But the reconnected edges pull their nodes together, and to a lower rate also the neighbors of them.
But for not so well structured graphs the Laplace layout is sometimes not very satisfying. See e.g. the layout in figure 5.4 for $\rho = 0$. The underlying graph has 40 nodes and 99 edges. The minimum degree is 1, the maximum 22, the mean is 4.9 and the standard deviation is about 3.7. In the layout some loosly connected nodes are overly separated from the rest, which build a dense cluster. Most adjacent nodes are close together. But the nodes are not well distributed in the drawing area and the layout is not very significant. The reason for this is that the weighted edge lengths are minimized the most, if the

low weighted edges of some less connected nodes become large and the rest of the edges is very short. Well-structured graphs are mostly immune to this effect, since they have no loosly connected nodes. Unfortunately real world graphs are often like our test graph less structured and symmetric. To be able to lay out a larger class of graphs reasonable, modifications of E_L were introduced, which manipulate the influence of the degrees. Two such methods will be covered in the next sections.

5.3 Relaxed Laplace Layout

Let $G = (V, E, \omega)$ be a graph with relaxed Laplace matrix L_ρ. If not mentioned otherwise is $\rho \in [0, 1]$. In [BW] the idea of a relaxed Laplace layout appeared the first time. We give a modified definition:

Definition 5.7 (Relaxed Laplace Layout)
Assume that the eigenvalues of L_ρ are $\lambda_1^{L_\rho} \leq ... \leq \lambda_n^{L_\rho}$ with correspondending orthonormal eigenvectors $y^{(1)}, ..., y^{(n)} \in \mathbb{R}^n$. For a p-dimensioal *relaxed Laplace layout* of G consider the set $y^{(1)}, ..., y^{(p+1)}$. If all weights are nonnegative, we remove $y^{(1)}$ from this set. If there are negative weights, we remove the eigenvector, for which the term

$$\left|\left| y^{(i)} - \frac{1}{\sqrt{n}} \mathbf{1} \right|\right| \quad , 1 \leq i \leq p+1 \quad ,$$

becomes minimal. The remaining p eigenvectors are then the axis vectors of the layout.

The relaxed Laplace layout is equal to the Laplace layout, if the $\rho = 0$ or the graph is regular. In this cases everthing from this section carries over to the Laplace layout.
Let us consider the relaxed Laplace matrix as matrix that varies in a factor ρ. Suppose first that all weights are nonnegative. There is one unique eigenvalue function $f_1(\rho)$ with $f_1(0) = 0$. The corresponding eigenvector is $\mathbf{1}$ for $\rho = 0$. The eigenvalue function f_1 is simple, there are no intersections between f_1 and other eigenvalue functions and therefore its eigenvector is continuous. If ρ is in the important interval $]0, 1]$, then the eigenvector is close to $\mathbf{1}$. Hence we exclude this vector from the layout, otherwise it is not significant, see figure 5.5.
If we allow negative weights, then there may be some eigenvalues smaller than $f_1(\rho)$, perhaps as much as the dimension of the layout. To decide which eigenvector to exclude we compare eigenvectors with $\mathbf{1}$, as in the definition. We could also consider the Rayleigh-

Ritz coefficient $\frac{\mathbf{1}^T L_\rho \mathbf{1}}{n}$ and the residuum $r(L_\rho, \frac{1}{\sqrt{n}}\mathbf{1})$:

$$\begin{aligned}\frac{\mathbf{1}^T L_\rho \mathbf{1}}{n} &= -\rho \frac{\sum_{i=1}^n d_i}{n} \\ &= -\rho\,\overline{d} \quad , \text{ where } \overline{d} \text{ is the mean of all degrees.}\end{aligned}$$

$$\begin{aligned}r(L_\rho, \frac{1}{\sqrt{n}}\mathbf{1}) &= \left|\left| L_\rho \frac{1}{\sqrt{n}}\mathbf{1} - \left(\frac{\mathbf{1}^T L_\rho \mathbf{1}}{n}\right)\frac{1}{\sqrt{n}}\mathbf{1} \right|\right| \\ &= \left|\left| \frac{-\rho}{\sqrt{n}} \begin{pmatrix} \vdots \\ d_i \\ \vdots \end{pmatrix} - \frac{-\rho\,\overline{d}}{\sqrt{n}}\mathbf{1} \right|\right| \\ &= \rho \sqrt{\frac{\sum_{i=1}^n \left(d_i - \overline{d}\right)^2}{n}} \\ &= \rho\,\sigma(D\mathbf{1}) \quad .\end{aligned}$$

The Rayleigh-Ritz coefficient is equal to ρ multiplicated with the mean of all degrees. The residuum is equal to ρ multiplicated with the standard deviation over all degrees. Suppose λ is the eigenvalue of L_ρ, which is closest to the coefficient, and has x as corresponding eigenvector. From the theorems 3.16 and 3.17 we know that the residuum (together with the gap γ) bounds the distance of the Rayleigh-Ritz coefficient of being λ and the distance of $\mathbf{1}$ being the corresponding eigenvector:

$$\begin{aligned}\left|\lambda + \rho\overline{d}\right| &\leq \rho\,\sigma(D\mathbf{1}) \\ |\sin\angle(x, \mathbf{1})| &\leq \rho\,\sigma(D\mathbf{1})/\gamma \quad .\end{aligned}$$

The residuum is a tight bound: It is zero, iff $\rho = 0$ or the graph is regular, i.e. iff $\mathbf{1}$ is an eigenvector. So alternatively we could exclude the eigenvector x of λ, which is the eigenvalue closest to the Rayleigh-Ritz coefficient $\frac{\mathbf{1}^T L_\rho \mathbf{1}}{n} = -\rho\,\overline{d}$. But there may be constellations, where because of intersections other eigenvalues than $f_1(\rho)$ are closer to $-\rho\,\overline{d}$. This may coincide with a small gap γ. The wrong vector would be excluded. In contrast the definition above ensures that the for the layout worst eigenvector is taken out.

The eigenvectors of L_ρ are continuous, unless there are intersections in the eigenvalue functions. Even then there is not necessary a discontinuity. So let us assume that the eigenvectors of L_ρ are continuous functions in ρ. In lemma 4.9 we showed that iff $\mathbf{1}$ is an eigenvalue of L_ρ, then the Laplace and the relaxed Laplace layout are equal. If x is close to $\mathbf{1}$, all other eigenvectors of L_ρ are close to eigenvectors of L, too. Under this assumption $\rho\,\sigma(D\mathbf{1})/\gamma$ is also a tight upper bound for the dissimilarity of a relaxed Laplace layout

and the Laplace layout.
The relaxed Laplace layout can be formulated as an optimization problem:

Theorem 5.8
Given is a graph $G = (V, E, \omega)$. If the weights G are nonnegative, a p-dimensional relaxed Laplace layout l are the vectors $y^{(2)}, ..., y^{(p+1)}$ of the solution of the following constraint minimization problem. If there are negative weights, a p-dimensional relaxed Laplace l are the p vectors of the solution, which are the least similar to $\mathbf{1}$.

$$\min_{y^{(1)},...,y^{(p+1)}} E_L(p) - \rho \sum_{i=1}^{p} (y^{(i)})^T D y^{(i)} \;=$$

$$\min_{y^{(1)},...,y^{(p+1)}} \sum_{\substack{(i,j)\in E \\ i\neq j}} \omega_{ij} \Big[|| \, l^{(i)} - l^{(j)} ||^2 - \rho \Big(|| \, l^{(i)} ||^2 + || \, l^{(j)} ||^2 \Big) \Big] - \rho \sum_{i=1}^{n} \omega_{ii} || \, l^{(i)} ||^2$$

$$\text{subject to: } (y^{(i)})^T y^{(j)} = \delta_{ij} \,, \quad 1 \leq i < j \leq p+1 \quad .$$

The theorem follows with the Rayleigh-Ritz eigenvalue characterizaton and lemma 4.8. The energy function is changed in the relaxed Laplace layout. A penalty term is introduced. The weighted edge lenghts are still minimized, but additionally the weighted distances from the origin are maximized depending on ρ. On the one hand nodes with a high (positive) degree are forced to move away from the origin, because they have much impact on the energy function. On the other hand nodes with smaller degrees are forced to move between the nodes with high degrees to keep the edge lenghts short. Therefore the high degree nodes are often ordered circularly. With an appropriate ρ the typical situation for Laplace layouts can be avoided, in which there is a dense cluster of nodes together with some loosly connected ones at the boundary. In figure 5.4 this is illustrated. Also the continuousity of the layouts is visible. In graphs without self-loops the symmetries from the Laplace layout are preserved, since the weighted edge lengths are still minimized. See figure 5.6 as an example.
Self-loops are now included in the energy function. They move nodes away from the origin, if they are positive, and towards the origin, if they are negative. See for example figure 5.1. There are two graphs in a relaxed Laplace layout with $\rho = 1$. The right one is the left one with two self-loops added. Note that the left one is regular and so any ρ could have been chosen for it. If we set $\rho = 0$, the layouts are equal, since in a Laplace layout self-loops have no influence.

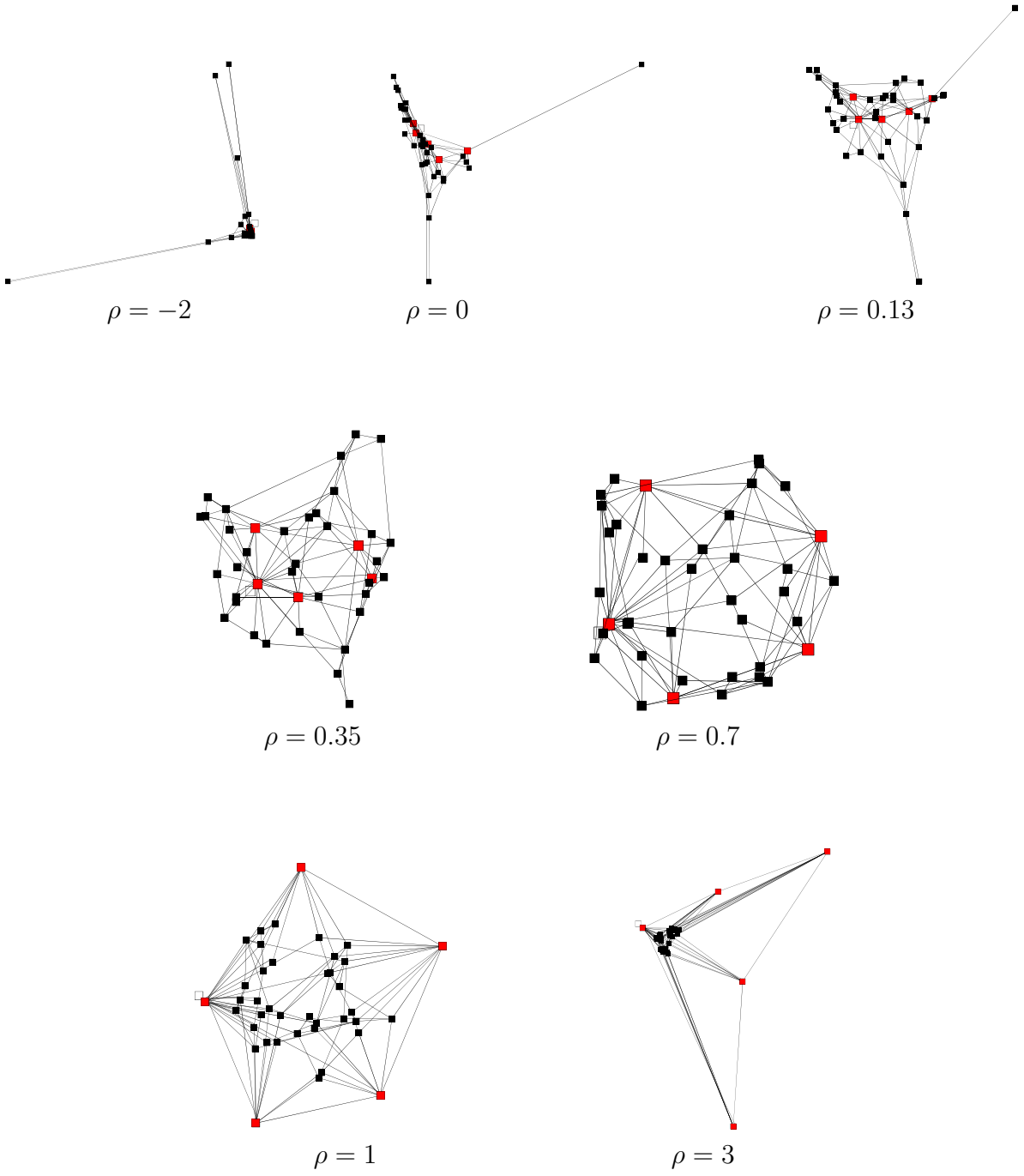


Figure 5.4: Relaxed Laplace layouts of a graph. The red nodes have the highest degrees.

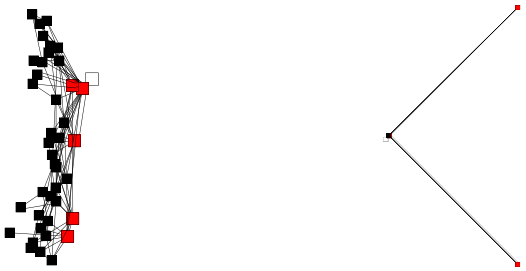


Figure 5.5: Graph from figure 5.4. On the left drawn using the eigenvectors of the smallest and the second-smallest eigenvalue of L_ρ with $\rho = 0.35$. On the right relaxed Laplace layout with $\rho = -1000$.

Let a graph be given, that is not regular. We show, that valuable layouts can mostly be found for $\rho \in [0, 1]$. If $\rho = 0$, the overall edge lengths are minimized. Since $L_\rho = (1 - \rho)D - A$, the influence of D is minimized for $\rho \in]0, 1]$ and vanishes for $\rho = 1$, whereas the influence of $-A$ is increased. The eigenvectors of D are unit vectors or linear combinations of unit vectors, if the eigenvalues are multiple. So the degree matrix emphasizes in each dimension of a relaxed Laplace layout a few nodes (1-entries in the eigenvectors) and tries to set all other node coordinates zero (0-entries in the eigenvector). The energy function for $L_\rho = -A$ is

$$-2 \sum_{\substack{(i,j)\in E \\ i\neq j}} \omega_{ij} \left(l^{(i)}\right)^T l^{(j)} - \sum_{i=1}^{n} \omega_{ii} || \, l^{(i)} ||^2 \quad .$$

To minimize this term under the given constraints, higher degree nodes are forced to be more distant from the origin than lower degree nodes. Edge lengths are secondary.
The matrices D and $-A$ control each other: On the one hand the layout consists of a few separate nodes and one dense cluster, if D becomes dominant in L_ρ. In figure 5.5 we set $\rho = -1000$ for the graph from figure 5.4. The layout is formed like a "L". The same happens for $\rho = 1000$. If $|\rho|$ decreases, the "L"-form begins to converge to the usual spectral layout form. But some graphs are still similar to the degree matrix form for $\rho = 0$, like our one. On the other hand the degree matrix D treats most nodes equally, like 0-entries in its eigenvector. For $\rho \in]0, 1]$ it moderates the growing influence of $-A$, which emphasizes nodes according to their degrees. If $\rho \in [1, 2]$, the influence of D is measured in ρ the same as for $\rho \in [0, 1]$. But D does not act as moderator any more, since its sign has changed in L_ρ. The the above described effects of $-A$ are even intensified and the layout is mostly bad. For $\rho < 0$ and $\rho > 2$ the degree matrix becomes dominant anyway. If the graph is highly structured or even regular, the layouts are well balanced for $\rho = 0$.

These graphs have no loosly connected nodes, that could easily be pushed to the boundary. It is optimal for them just to minimize over the weighted edge lengths.

We now examine the deviation of node i in the dimension k from being placed in the weighted centroid $c_\omega(i)$:

$$\begin{aligned} x_i^{(k)} - c_\omega(i) &= x_i^{(k)} - \frac{\sum_{j \in N(i)} \omega_{ij} x_j^{(k)}}{d_i - \omega_{ii}} \\ &= \frac{d_i x_i^{(k)} - \omega_{ii} x_i^{(k)} - \sum_{j \in N(i)} \omega_{ij} x_j^{(k)}}{d_i - \omega_{ii}} \\ &= \frac{\lambda^{L_\rho} x_i^{(k)} + \rho d_i x_i^{(k)}}{d_i - \omega_{ii}} \\ &= \left(\frac{\lambda^{L_\rho} + \rho d_i}{d_i - \omega_{ii}} \right) x_i^{(k)} \quad . \end{aligned}$$

The deviation depends on the degree, on ρ and on the eigenvalue λ^{L_ρ} corresponding to the axis vector $x^{(k)}$. If $\rho = 0$, then deviation becomes larger, if the degree becomes smaller. This is also a description of the situation, where there is one dense cluster and some low-degree nodes at the boundary in some Laplace layouts. Assume now for simplicity that the underlying graph has no self-loops. Then the deviation can be written as

$$\left(\frac{\lambda^{L_\rho}}{d_i} + \rho \right) x_i^{(k)} \quad .$$

Again nodes with the same degree have the same percentage deviation, but it is difficult to predict from this term, which deviation a certain class of nodes has. The eigenvalue may be positive or negative and the fraction may lie between $-\rho$ and 0 or not. The eigenvalue and ρ depend on each other. By looking at the energy function we already found out that with a growing ρ the high-degree nodes move away from the origin and the low degree nodes move between them. If there are major differences in the degrees of a graph, then the relaxed Laplace layout can be very similar to a barycentric layout. The high-degree nodes have the role of the nodes fixed in advance. They build a circle around the low-degree nodes. These are placed very close to their $c_\omega(i)$, in opposite to the high-degree nodes. The layout for $\rho = 1$ in figure 5.4 is an example.

Which ρ is now the optimal value for which graph? The first thing to state is that an appropriate ρ depends on the graph, that should be laid out. In figure 5.4 the layout is for $0.35 \leq \rho \leq 0.7$ acceptable. The edges are not too long and the nodes are well distributed. In figure 5.6 there is another layout with $\rho = 0.7$. It is symmetric and circularly, but the edge lengths have grown, since the distance of the nodes from the origin

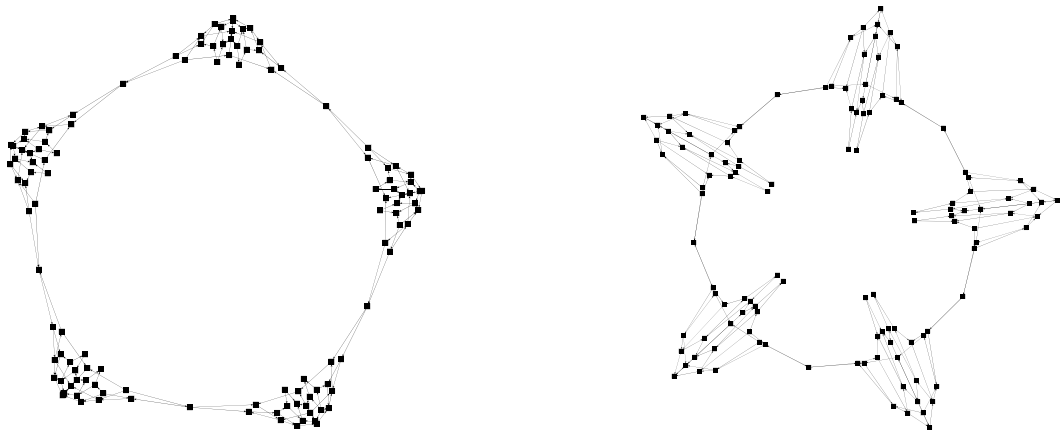


Figure 5.6: Wheel-graph from figure 5.3 in a relaxed Laplace layouts.

begins to correspond with the degree. The layout for $\rho = 0.17$ is better and for $\rho = 0$ (figure 5.3 on the left) is the best. Edge lengths are shorter and more of the grid structure is apparent. Another behaviour we observed for the small world graph of figure 5.3 on the right. It is very close to be regular and so for $0 \leq \rho \leq 1$ there are no differences visible in the layouts.

For less structured graphs ρ should be large enough to stretch the cluster and to move outliers back to the rest. But the layout should not be barycentric-like as for $\rho = 1$ in figure 5.4. The amount a graph is stretched for a certain ρ depends on how good it is connected. The higher the degrees of the nodes, the more the overall edge lengths grow, if they are moved, and the higher ρ must be chosen. In contrast more structured graphs often already have nice Laplace layouts. The relaxation factor should be close to 0 for them.

Let an unweighted graph $G = (V, E)$ without self-loops be given. Then in our experience the following easy to compute heuristic generates reasonable layouts:

$$\rho_h := \sqrt{\frac{\frac{1}{n}\sum_{i=1}^{n} d_i}{n-1}} = \sqrt{\frac{|V|}{n(n-1)}} \quad .$$

The number of (directed) edges in G is divided by the number of possible edges in G and the root is taken. For all graphs yields $\rho_h \in [0, 1]$. The heuristic grows in the average degree of the nodes. For well structured graphs we assume that their average degree is low or they are close to be regular, so that ρ_h has less effect. An example for such a graph is figure 5.6. We got $\rho_h \approx 0.17$, which is close enough to 0 here. For the not so well

structured graph from figure 5.4 we got $\rho \approx 0.35$ and thereby also a satisfactory layout. A more general approach also valid for weighted graphs would be even more satifying, however.

5.4 Generalized Laplace Layout

Due to Koren [Ko] is the following spectral graph drawing technique, that uses generalized eigenvectors as axis vectors. In our version self-loops and negative weights are allowed, as long as all degrees remain positive. All results from Koren also yield for our version. Let $G = (V, E, \omega)$ be a graph with with positive degrees. Let L be its Laplace matrix and D its degree matrix. The generalized eigenpairs of (L, D) are by construction the eigenpairs of the generalized Laplace matrix L_G (see section 4.5). The trivial eigenvector $\mathbf{1}$ should be exlcuded again. Then:

Definition 5.9
Let denote $\lambda_1^{L_G} \leq ... \leq \lambda_n^{L_G}$ the eigenvectors of L_G with correspondending eigenvectors $y^{(1)}, ..., y^{(n)} \in \mathbb{R}^n$. The p-dimensional *generalized Laplace layout* of G is defined by p axis vectors $x^{(i)}$:

$$x^{(i)} := y^{(i+1)} \quad , 1 \leq i \leq p \quad .$$

We already pointed out in section 4.5, that the generalized eigenvectors of (L_ρ, D) are equal to the generalized eigenvalues of (L, D). Again the layout can be formulated as as an solution of an optimization problem:

Theorem 5.10
A p-dimensional generalized Laplace layout l of a graph is the solution of the following

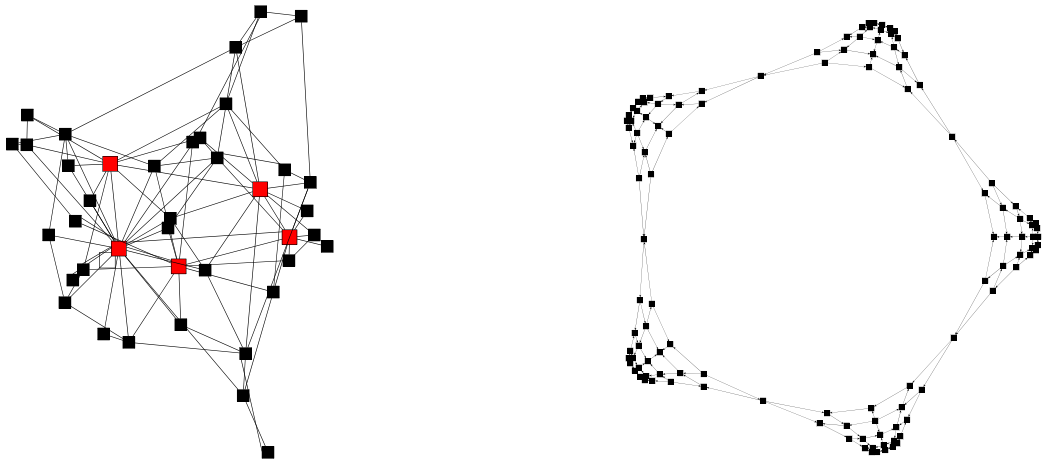


Figure 5.7: Generalized layouts. On the left graph from figure 5.4, on the rigth wheel graph from figure 5.6.

constraint minimization problem:

$$\min_{x^{(1)},\ldots,x^{(p)}} \frac{E_L(p)}{\sum_{i=1}^{p} \left(x^{(i)}\right)^T D x^{(i)}} = \qquad (*)$$

$$\min_{x^{(1)},\ldots,x^{(p)}} \sum_{(i,j)\in E} \omega_{ij} \left(\frac{1}{d_i} l^{(i)} - \frac{1}{d_j} l^{(j)}\right) \left(l^{(i)} - l^{(j)}\right)$$

$$\text{subject to: } (x^{(i)})^T D x^{(j)} = \delta_{ij}\,, \quad 1 \le i < j \le p$$
$$\mathbf{1}^T D x^{(i)} = 0\,, \quad 1 \le i \le p \quad .$$

The theorem is obviously a direct consequence of theorem 4.13 and the Rayleigh-Ritz theorem 3.3 for generalized eigenproblems. For $E_L(p)$ the same penalty term is introduced as in the relaxed layout, but here by division. The effect is very similar. In $(*)$ the numerator, the overall weighted edge length, is minimized, as in the Laplace layout. And with the denominator also the same term is maximized. The highest degree nodes are forced to move away from the origin. Often they are ordered circularly. To keep edge lengths short lower degree nodes are scattered between the higher degree nodes. The combination of these two goals helps to prevent the situation, in which nodes with lower degrees are overly separated from the rest. For most graphs, structured or unstructured, the results are reasonable. There are no outliers and the nodes are well distrubuted. In figure 5.7 are two examples. Self-loops have the same effect as in the relaxed Laplace layout. Positive ones move the node away from the origin, negative ones towards the

origin.
We state now two characterizations of the generalized Laplace layout frome Koren [Ko, p. 11-15]. The difference of L_G and L is, that in L_G the nodes are weighted by their degrees, i.e. that every entry in row i is multiplied by $\frac{1}{d_i}$. Edge lengths in the generalized layout therefore depend on the relative importance of their corresponding nodes:

> "Consider two edges e_1 and e_2. Edge e_1 is of weight 1, connecting two nodes, each of which is degree 10. Edge e_2 is of weight 10, connecting two nodes, each of which is degree 100. In the Laplacian matrix, the entries corresponding to e_2 are ten times larger than those corresponding to e_1. Hence we expect the drawing obtained by the eigenvectors of the Laplacian, to make the edge e_2 much shorter than e_1 (here, we do not consider the effect of other nodes that may change the lengths of both edges). However, for the ...(matrix L_G)..., the entries corresponding to these two edges are the same, hence we treat them similarly and expect to get the same length for both edges. This reflects the fact that the *relative* importance of these two edges is the same, i.e. $\frac{1}{10}$."

It is further shown, that the generalized eigenvalue λ^{L_G} is also a measure for the deviation of node i in the layout dimension k from being placed in the weighted centroid $c_\omega(i)$:

$$\begin{aligned} x_i^{(k)} - c_\omega(i) &= x_i^{(k)} - \frac{\sum_{j \in N(i)} \omega_{ij} x_j^{(k)}}{d_i - \omega_{ii}} \\ &= \frac{d_i x_i^{(k)} - \omega_{ii} x_i^{(k)} - \sum_{j \in N(i)} \omega_{ij} x_j^{(k)}}{d_i - \omega_{ii}} \\ &= \left(\frac{\lambda^{L_G} d_i}{d_i - \omega_{ii}} \right) x_i^{(k)} \end{aligned}$$

The smaller the eigenvalue gets, the smaller are the deviations from c_ω. This confirms the choice of the eigenvectors of the smallest non-zero eigenvalues as axis vectors. Remarkable is, that every node has an equal percentage deviation of c_ω, only depending on self-loops. Therefore are in generalized layouts neither loosly connected nodes at the boundary nor situations similar to barycentric layouts as in the relaxed layout. The distibution of the nodes is always well balanced.

The relaxed and the generalized Laplace layouts are very similar by our observation. They are both equal to the Laplace layout for regular graphs. Their energy functions modifiy the original energy function $E_L(p)$ of the Laplace layout with the same penalty term $\sum_{i=1}^{p} \left(x^{(i)}\right)^T D x^{(i)}$ - by subtraction or division, respectively. In fact there is for all

graphs with positive degrees a relaxation factor ρ such that the energy functions are equal:

$$\begin{aligned} \frac{\sum_{i=1}^{p} \left(x^{(i)}\right)^T L x^{(i)}}{\sum_{i=1}^{p} \left(x^{(i)}\right)^T D x^{(i)}} &= \sum_{i=1}^{p} \left(x^{(i)}\right)^T L x^{(i)} - \rho \sum_{i=1}^{p} \left(x^{(i)}\right)^T D x^{(i)} \\ \Leftrightarrow \quad \rho &= \frac{\sum_{i=1}^{p} \left(x^{(i)}\right)^T L x^{(i)}}{\sum_{i=1}^{p} \left(x^{(i)}\right)^T D x^{(i)}} \left(1 - \frac{1}{\sum_{i=1}^{p} \left(x^{(i)}\right)^T D x^{(i)}}\right) \quad . \end{aligned}$$

For the computation of ρ we need the generalized layout of the underlying graph, hence this a theoretical result. It is an open question, if in general also the layouts are equal for this ρ or if it is prevented by the constraints in the optimization problems. It would be a necessary condition that $D\mathbf{1}$ is an eigenvector of L_ρ, since the eigenvectors of L_ρ were D-orthogonal to $\mathbf{1}$. Anyway there seem to be certain values of ρ for each graph, so that both layouts look nearly the same. The heuristic for ρ from the last section matches these values often, e.g. for the graphs from figure 5.7. The eigenvector to the smallest eigenvalue is then very close to $D\mathbf{1}$, see the x-dimension of the left graph in figure 5.5.

Chapter 6

Implementation

For a p-dimensional spectral layout we have to compute $p+1$ eigenvectors of a graph related matrix. As most authors we decided to implement for this the orthogonal iteration, a multidimensional power iteration. The algorithm alone is slower than the state-of-the-art techniques. But it is the base for many acceleration methods from numerical as well as from graph theorectial contexts. We used it for flexibility reasons. It is easy to implement and its correctness is proven. We were able to retrace convergence anormalities. Aside from that it works very reliable. With some modifications it provides us generalized eigenvalues and fast, well-looking results in dynamic graph drawing. For 2- or 3-dimensional layouts of graphs with less than 200 nodes the computation time was not too slow.
An eigensolver software, that has the function to compute the topmost few eigenpairs, could be used alternatively.
We used the java programming language Java™ 2 SDK, Standard Edition, Version 1.4.2. To have a graphical interface for manipulating graphs, we integrated our algorithms in the yEd[1] graph editor, version 2.2, using the yFiles class library, version 2.2.

6.1 A Spectral Layout Algorithm

To introduce the algorithm we first need some more theoretical background:

[1] yEd and yFiles are products of the company yWorks. For more information see their hompage `http://www.yWorks.com`

Theorem 6.1 (QR-Factorization)
If $A \in \mathbb{R}^{n\times n}$, there is a unitary matrix $Q \in \mathbb{R}^{n\times n}$ and an upper triangular matrix $R \in \mathbb{R}^{n\times n}$ such that

$$A = QR \quad .$$

If A is nonsingular, then R maybe chosen so that all its diagonal entries are positive, and in this event, the factors Q and R are both unique.

A constructive proof is stated in [HoJo, theorem 2.6.1]. There is the Gram-Schmidt orthonormalization process used to compute Q. It's not difficult to complete this process so that also R is computed. But computing R will not be necessary.

Theorem 6.2 (Gram-Schmidt orthonormalization process)
Given are m vectors $a^{(1)}, ..., a^{(m)} \in \mathbb{R}^n$, $m \leq n$. Then the following process provides a orthonormal set of vectors $q^{(1)}, ..., q^{(m)} \in \mathbb{R}^n$ of the same span, which are orthonormal unless they are equal $\mathbf{0}$:

If $a^{(1)} = \mathbf{0}$ set $q^{(1)} = \mathbf{0}$. Otherwise set $q^{(1)} = \dfrac{a^{(1)}}{||\, a^{(1)}||}$.

For each $j = 2, 3, ..., m$:

$$q^{(j)} = a^{(j)} - \sum_{i=1}^{j-1} \left((q^{(i)})^T a^{(j)} \right) q^{(i)}$$

If $q^{(j)} \neq \mathbf{0}$ set $q^{(j)} = \dfrac{q^{(j)}}{||\, q^{(j)}||}$.

The vector $q^{(j)}$ is equal $\mathbf{0}$, iff $a^{(j)}$ is a linear combination of $a^{(1)}, ..., a^{(j-1)}$.

Eigenvalue algorithms are always iterative, because an exact computation is too expensive. We use an algorithm called *orthogonal* or *simultaneous iteration* in literature. We show now, that this algorithm is closely linked to the *QR-algorithm*, for which there is a better theoretical basis. They are both stated in algorithm 1 in a version with a start approximation Q. The matrix Q will later on be used to speedup and direct the convergence.

Lemma 6.3
Given is a matrix $A \in \mathbb{R}^{n\times n}$. With the same notation as in algorithm 1 yields:

Algorithm 1 Two Eigensolvers

Given are $A \in \mathbb{R}^n$, $Q \in \mathbb{R}^n$ unitary.

procedure ORTHOGONAL ITERATION
  $\widehat{Q}_0 = Q$
  for $k = 1, ...$ **do**
    $\widehat{Q}_k \widehat{R}_k = A\widehat{Q}_{k-1}$ ▷ QR-factorization
  end for
  return $\widehat{Q}_k$
end procedure

procedure QR-ALGORITHM
  $Q_0 = Q$
  $A_0 = Q_0^T A Q_0$
  for $k = 1, ...$ **do**
    $Q_k R_k = A_{k-1}$ ▷ QR-factorization
    $A_k = R_k Q_k$
  end for
  return A_k
end procedure

- A_k is nonsingular, iff A is nonsingular.
- A_k is symmetric, iff A is symmetric.
- A_k has the same eigenvalues as A.

Proof:
The assertion stems from

$$A_k = R_k Q_k = Q_k^T Q_k R_k Q_k = Q_k^T A_{k-1} Q_k = ... = (Q_0 Q_1 \cdots Q_k)^T A Q_0 Q_1 \cdots Q_k \quad .$$

□

Lemma 6.4
Given is a nonsingular matrix $A \in \mathbb{R}^{n \times n}$. With the same notation as in algorithm 1 yields:

$$A_k = \widehat{Q}_k^T A \widehat{Q}_k \quad .$$

Proof:
We proof this by induction. Obviously holds $A_0 = \widehat{Q}_0^T A \widehat{Q}_0$. Now let us assume that $A_k = \widehat{Q}_k^T A \widehat{Q}_k$. Since $A\widehat{Q}_k = \widehat{Q}_{k+1} \widehat{R}_{k+1}$ we have

$$A_k = \widehat{Q}_k^T \widehat{Q}_{k+1} \widehat{R}_{k+1} = Q_{k+1} R_{k+1} \quad .$$

The matrix $\widehat{Q}_k^T \widehat{Q}_{k+1}$ is unitary, bcause $\widehat{Q}_k^T \widehat{Q}_{k+1} \left(\widehat{Q}_k^T \widehat{Q}_{k+1} \right)^T = I$. So both $\widehat{Q}_k^T \widehat{Q}_{k+1} \widehat{R}_{k+1}$ and $Q_{k+1} R_{k+1}$ are QR-factorizations of A_k. From lemma 6.3 we know that A_k is nonsingular. We choose then the factorization to be unique, i.e. $\widehat{Q}_k^T \widehat{Q}_{k+1} = Q_{k+1}$ and $\widehat{R}_{k+1} = R_{k+1}$. Then yields

$$\begin{aligned} A_{k+1} &= R_{k+1} Q_{k+1} \\ &= \widehat{R}_{k+1} \widehat{Q}_k^T \widehat{Q}_{k+1} \\ &= \widehat{Q}_{k+1}^T \widehat{Q}_{k+1} \widehat{R}_{k+1} \widehat{Q}_k^T \widehat{Q}_{k+1} \\ &= \widehat{Q}_{k+1}^T A \widehat{Q}_k \widehat{Q}_k^T \widehat{Q}_{k+1} \\ &= \widehat{Q}_{k+1}^T A \widehat{Q}_{k+1} \quad . \end{aligned}$$

□

Wilkinson states a formal proof of the convergence of the QR-algorithm in [Wi, p. 515-521]. Unfortunatly there is no theorem-like proposition and the proof is very detailed. So we give a brief sketch of it as a reading guide and present some important results. We focus on real symmetric, positive definite matrices. For them the convergence statements are more accurate and the proof is easier. Our QR-algorithm differs a bit from Wilkinson's: He starts with $A_0 = A$ instead of $A_0 = Q_0^T A Q_0$, Q_0 unitary. But because in both cases A_0 has the same eigenvalues and A_0 remains symmetric if A is symmetric, this does not affect the proof.

Theorem 6.5 (Convergence of the QR-Algorithm)
Given is a symmetric, positive definite matrix $A \in \mathbb{R}^{n \times n}$. Let $\lambda_1 \leq .. \leq \lambda_n$ denote the eigenvalues of A. With the same notation as in algorithm 1 yields:

- $A_k \to D$ as $i \to \infty$, with $D = (d_{ii})$ diagonalmatrix with the eigenvalues of A on the diagonal. This is still true, if A has eigenvalues of equal modulus.

- Let $A_0 = KTK^T$ be a Schur decomposition of A_0 with $T = (t_{ii})$ diagonal and $t_{ii} = \lambda_{n-i+1}$. I.e. t_{11} is the largest eigenvalue, t_{22} the second-largest and so on. Suppose all leading principal minors of K^T are nonzero. Then the d_{ii} is also λ_{n-i+1}. Otherwise the diagonal entries of D are a permutation of the diagonal entries of T.

- The convergence depends on

$$\max_{\substack{\lambda_i \neq \lambda_{i+1} \\ 1 \leq i \leq n-1}} \left(\frac{\lambda_i}{\lambda_{i+1}} \right)^k \to 0 \quad .$$

Proof (sketched):
Since A_0 is positive definite and therefore nonsingular, both A_k and $(A_0)^k$ are positive definite and nonsingular. We have seen in lemma 6.3, that A_k is symmetric if A_0 is. The same holds for $(A_0)^k$. We define

$$\begin{aligned} P_k &= Q_1 Q_2 \cdots Q_k \\ U_k &= R_k R_{k-1} \cdots R_1 \end{aligned}$$

It can be shown that

$$P_k U_k = (A_0)^k$$

P_k is unitary and U_k is an upper triangular. We can choose the R_k to have postive diagonal entries and hence to be unique, because all A_k are nonsingular. It follows that U_k has positive diagonal entries and therefore P_kU_k is the unique QR-decomposition of $(A_0)^k$. Let

$$(A_0)^k \quad = \quad KT^kK^T \quad , \qquad (*)$$

be the Schur decompositions of $(A_0)^k$ with K and T as in the assertion. Under the assumption, that all leading principal minors of K^T are nonzero, Wilkinson factorizes the Schur decomposition into an unitary and an right upper matrix. Here we leave out a larger part, since the decomposition is not very intuitive and less important for the general understanding of the proof. The right upper matrix has positive diagonal entries. Together with the unitary matrix they are also an unique QR-decomposition of $(A_0)^k$. The unitary matrix converges to K and so must P_k. From the proof of lemma 1 we know, that $A_k = (P_k)^T A_0 P_k$. Therefore converges A_k to T. It is essential for convergence, that the eigenvalues in T are ordered as in the assertion!
Wilkinsons further points out, that eigenvalues of A_0, which are equal, do not prevent the convergence. From the QR-decomposition of KT^kK^T can be concluded that convergence depends on

$$\max_{\substack{\lambda_i \neq \lambda_j \\ 1 \leq i,j \leq n}} \left(\frac{\lambda_i}{\lambda_j}\right)^k = \max_{\substack{\lambda_i \neq \lambda_{i+1} \\ 1 \leq i \leq n-1}} \left(\frac{\lambda_i}{\lambda_{i+1}}\right)^k \to 0 \quad .$$

It is necessary for the QR-decomposition of KT^kK^T to build a LU-factorization of K^T. Since K^T is nonsingular, there is such a factorization, iff all leading principal minors of K^T are zero. Suppose K^T has a leading principal minor equal zero. Then there exists a permutation matrix P, so that all principal minors of PK^T are nonzero [HoJo, pp. 158-163]. The matrix P permutes also the QR-decomposition and ultimately the entries of D.
As mentioned before all convergence statements for A_0 carry over to A. □

Theorem 6.6 (Convergence of the Orthogonal Iteration)
Given is a symmetric, positive definite matrix $A \in \mathbb{R}^{n \times n}$. With the same notation as in algorithm 1 yields: The column vectors of $\widehat{Q}_k$ converge to the eigenvectors corresponding to eigenvalues on the diagonal of A_k.

The convergence of the orthogonal iteration follows directly from lemma 6.4 and the convergence theorem of the QR-algorithm. Another, less general theorem on the convergence

of the orthogonal iteration is stated in [GvL, theorem 8.2.2].
The orthogonal iteration has several advantages. It provides us directly the eigenvalues. The matrix R of the QR-factorization needs not to be generated. To compute only p eigenvectors corresponding to the p top eigenvalues of a matrix, we can choose $\widehat{Q}_k$ to be in $\mathbb{R}^{n\times p}$. The latter follows directly from the Gram-Schmidt factorization. The convergence depends then on

$$\max_{\substack{\lambda_i \neq \lambda_{i+1} \\ 1 \leq i \leq p-1}} \left(\frac{\lambda_i}{\lambda_{i+1}}\right)^k \to 0 \quad .$$

The costs per iteration for computing p eigenvectors are in $O(|E|)$, where $|E|$ is the number of edges.
The most common way to describe the convergence of the orthogonal iteration extends the proof of the *power iteration*: Given is a symmetric, positive definite matrix $A \in \mathbb{R}^{n\times n}$ and a start vector $x \in \mathbb{R}^n$, $x \neq \mathbf{0}$. The vector x is a linear combination of an orthonormal basis of eigenvectors $q^{(i)}$ of A:

$$x = \sum_{i=1}^{n} c_i q^{(i)}\,, \quad c_i \in \mathbb{R},\, 1 \leq i \leq n\,.$$

To every $q^{(i)}$ correspondends an eigenvalue λ_i, $0 < \lambda_1 \leq ... \leq \lambda_n$. Then yields:

$$\begin{aligned} A^k x &= \sum_{i=1}^{n} \lambda_i^k c_i q^{(i)} \\ &= \lambda_n^k \left(c_n q^{(n)} + \sum_{i=1}^{n} \left(\frac{\lambda_i}{\lambda_n}\right)^k c_i q^{(i)} \right) \end{aligned}$$

So $A^k x$ converges to an eigenvector corresponding to the largest eigenvalue of A. After each multiplication with A the vector x should be normalized to avoid an extensive growth. This can be done, since the direction of the vector is not affect. This techique so far is called power iteration.
Suppose, we want to compute a second eigenvector. Let y be another iteration vector with

$$y = \sum_{i=1}^{n} d_i q^{(i)}\,, \quad d_i \in \mathbb{R},\, 1 \leq i \leq n$$

$$A^k y = \sum_{i=1}^{n} \lambda_i^k d_i q^{(i)}$$

The idea is to project y after each multiplication in a to x orthogonal subspace. As x converges to $q^{(n)}$, the subspace becomes orthogonal to $q^{(n)}$. The constant d_n gets close to zero. Then it is often assumed that y converges to $q^{(n-1)}$, the eigenvector to the largest remaining eigenvalue in that space. But the described method is equal to the orthogonal iteration with $p = 2$. In Wilkinson's proof is described, that under certain conditions y may converge to another eigenvector. In this case not only d_n becomes zero because of the projection, but also d_{n-1} and maybe some more. The subspace is orthogonal to $q^{(n)}$, $q^{(n-1)}$ and the other eigenvectors with a zero d_i. The iteration vector converges to the eigenvector of the maximum remaining eigenvalue. This can be extended to further iteration vectors. In the next section we will see some examples of this effect. Since the first iteration vector x is not projected into subspaces, it always converges to $q^{(n)}$, the eigenvector of the largest eigenvalue.

Our graph related matrices are all symmetric except the generalized Laplace matrix L_G. But we can compute their eigenvalues easily from the eigenvalues of a symmetric one, the normalized Laplace matrix L_N: (λ, u) is an eigenpair of L_N, iff $(\lambda, D^{-\frac{1}{2}}u)$ is an eigenpair of L_G (see lemma 4.16). By construction we obtain the generalized eigenvalues of (L, D) with the eigenvalues of L_G.
The first eigenvectors in $\widehat{Q}_k$ correspondend to the largest eigenvalues of A. To compute eigenvectors correspondending to the smallest few eigenvalues, we have to use the following shift:

Lemma 6.7
Given is a symmetric matrix $A \in \mathbb{R}^{n\times n}$ with upper eigenvalue bound $b \in \mathbb{R}$. Let $\epsilon > 0$. Then the iteration matrix

$$A_I := (b + \epsilon)I - A$$

is positive definite. The eigenvectors corresponding to A_I's top eigenvalues are the eigenvectors corresponding to A's smallest eigenvalues .

This is again a conclusion from lemma 2.3. What we left out so far are stop criteria. One possibility is to terminate the orthogonal iteration, if the change pro iteration in $\widehat{Q}_k$ is negligible. But if a matrix converges slow, the algorithm may terminate too early. In [BS, section 4.2] is illustrated, that the iterate quickly moves toward a subspace spanned by eigenvectors associated with the largest eigenvalues of the iterated matrix. Layouts built with such an iterate are mostly acceptable. But their energy functions could still be minimized more. So a better stop criterion is the distance of the column vectors q_k of $\widehat{Q}_k$

of being an eigenvector. A good measure is the residuum

$$r(A, q_k) = || Aq_k - \frac{(q_k)^T Aq_k}{(q_k)^T q_k} q_k ||^2 \quad ,$$

see theorem 3.17. To save costs the expensive matrix-vector multiplications should be avoided. The trick is to compute the residuum of step k-1 in step k of the orthogonal iteration. Then:

$$r(A, q_{k-1}) = || q_k - \left((q_{k-1})^T q_k\right) q_{k-1} ||^2 \quad .$$

This must be done after the matrix multiplication and before the orthogonalization. We propose to terminate the orthogonal iteration, if the maximal residuum over all column vectors of $\widehat{Q}_k$ is negligible or if a maximum of iterations is reached. The reason for the latter is, that there are some ill-conditioned matrices converging too slow, as it would make sense to wait for the correct termination. The algorithm halts with the trick one iteration too late, but the result is not worsened.
With this knowledge we can state now our p-dimensional spectral graph layouter as algorithm 2.

One simple method to increase the convergence speed is to improve the upper eigenvalue bound b. Obviously all eigenvalues of A_I and hence the term $\frac{\lambda_i}{\lambda_{i+1}}$, $\lambda_i \neq \lambda_{i+1}$, become smaller then. For this reason we intensively studied eigenvalue bounds in section 4.8. Using the upper bound of theorem 4.22 instead of the usual Gershgorin bound accelerated the convergence up to 50%, depending on the second-largest degree of the underlying graph. We should also set $\epsilon := 0$ in the shift of A_I. In practice this does not affect the convergence.
In the next section an alternative procedure ITERATE is introduced, that measures the convergence of each single iteration vector and not the maximal deviation from an eigenvector over all iteration vectors. The convergence speed of the procedure is therefore slightly better.
More effective is finding a good start approximation for Q_0. In the context of dynamic graph drawing we took already computed eigenvectors of very similar graphs and had fine results. See section 7 for more details. In [KCH] a similar technique is described: The original graph is successional approximated by a number of *coarse graphs*, every one of about half the size. The eigenvectors of the coarse graphs are then computed recursively, using a modification of the precessor's (lower-dimensional) eigenvectors as start vectors.
Koren suggests in [Ko, chapter 7] another possibility: He constructs for the graph a suitable lower-dimensional subspace. In this space the iteration is much faster. The result is then projected back in $\mathbb{R}^n$. Standard eigensolvers get problems by his experience, if a graph has more than 10^5 nodes. Then such especially for graph-related matrices designed techniques should in any case be preferred.

Algorithm 2 p-Dimensional Spectral Graph Layouter

Given is L, D, $L_\rho \in \mathbb{R}^{n\times n}$ and $Q_k := \left(q_k^{(1)}, ..., q_k^{(p+1)}\right) \in \mathbb{R}^{n\times(p+1)}$.
Let Q_0 be a unitary start approximation, δ the precision and $k_{\max}$ maximum of steps.

procedure RELAXED LAPLACE LAYOUT(L_ρ, Q_0, p)
  compute upper eigenvalue bound b of L_ρ
  $A_I = (b+\epsilon)I - L_\rho$
  ITERATE(A_I, Q_0, $p+1$)
  return $q_k^{(2)}, ..., q_k^{(p+1)}$ if all weights are nonnegative, otherwise Q_k without the vector most similar to $\mathbf{1}$
end procedure

procedure GENERALIZED LAPLACE LAYOUT (L, D, Q_0, p)
  $L_G = D^{-\frac{1}{2}} L D^{-\frac{1}{2}}$ ▷ All degrees must be positive here.
  compute upper eigenvalue bound b of L_ρ
  $A_I = (b+\epsilon)I - L_G$
  ITERATE(A_I, Q_0, $p+1$)
  $Q_k = D^{-\frac{1}{2}} Q_k$
  for $j = 1, ..., p$ **do**
    $q_k^{(j)} = \frac{q_k^{(j)}}{|| q_k^{(j)} ||}$
  end for
  return $q_k^{(2)}, ..., q_k^{(p+1)}$
end procedure

procedure ITERATE(A, Q_0, p)
  repeat
    $Q_k = AQ_{k-1}$
    for $j = 1, ..., p$ **do**
      $r(A, q_{k-1}^{(j)}) = || q_k^{(j)} - \left((q_{k-1}^{(j)})^T q_k^{(j)}\right) q_{k-1}^{(j)} ||^2$
      $q_k^{(j)} = q_k^{(j)} - \sum_{i=1}^{j-1} \left((q_k^{(i)})^T q_k^{(j)}\right) q_k^{(i)}$
      $q_k^{(j)} = \frac{q_k^{(j)}}{|| q_k^{(j)} ||}$
    end for
  until $\max_{1\le j\le p} r(A, q_{k-1}^{(j)}) \le \delta$ or $k > k_{\max}$
  return Q_k
end procedure

In graph drawing related literature there are several other algorithms described to obtain the smallest eigenvectors of a graph related matrix. Mostly they are power iteration (respectively orthogonal iteration) based and compute the eigenvectors of the Laplace matrix L. Koren [Ko] states a version, that computes the generalized eigenvalues of (L, D). The Gram-Schmidt orthonormalization is modified, since the eigenvectors are not orthogonal but D-orthogonal. In [BW] an algorithm can be found for the eigenvalues of L_ρ. The orthonormalization process is modified here, too: Instead of the eigenvector corresponding to the smallest eigenvalue, the iterate vectors are orthogonalized to $\mathbf{1}$. This vector is an eigenvector of L_ρ, iff $\rho = 0$ or the graph is regular (see section 4.4). Otherwise the algorithm does converge, too, since the eigenvector corresponding to the smallest eigenvalue is not very different from $\mathbf{1}$. But the eigenvectors are not exactly the result. The residuum cannot be used as convergence criterion. The optical deviation in the layouts is negligible, however. Our algorithm unifies these two approaches and has an exact convergence proposition. Similar effects as we describe in the next section can be observed for both of them.

6.2 Convergence Anormalities

We first examine the eigenvalue interchange effect. Given is a graph related symmetric matrix $A \in \mathbb{R}^{n\times n}$ and its iteration matrix $A_I = (b+\epsilon)I - A$, with $b \in \mathbb{R}$ an upper eigenvalue bound of A. The iteration matrix is positive definite and has the Schur decomposition $A_I = KTK^T$, $K \in \mathbb{R}^{n\times n}$ unitary and $T \in \mathbb{R}^{n\times n}$ diagonal. The columns of matrix K are the eigenvectors corresponding to the eigenvalues on the diagonal of T. The eigenvalues are ordered decreasing in T. As in the proof of the QR-algorithm shown the eigenvalue interchange effect occurs, if a leading principal minor of K^T is zero. It is difficult to predict such a behavior, since we know not much about K^T except that it is nonsingular and in general not symmetric. Before the iteration the eigenvectors are unknown, as a matter of fact.
With algorithm 2 we computed the smallest eigenvalues of L_ρ for several graphs. As initialization we used the first unit vectors. We stopped the algorithm, when the maximal residuum was smaller than 10^{-6}. Figure 6.1 shows four critical unweighted graphs[2] with their corresponding eigenvalue diagram on the right. The diagrams show the eigenvalues for $0 \leq \rho \leq 1$, denoted by the order, in which the algorithm provided them. The provided order of the second graph and the forth matches the eigenvalue order. For the first and the third graph there is a mismatch. Their K^T has a zero leading principal minor. We

[2]The graphs are drawn by hand in this section.

document this fact for the first graph and $\rho = 1$:

$$A_I = \begin{pmatrix} 6 & 0 & -1 & 0 & 0 \\ 0 & 6 & 0 & -1 & 0 \\ -1 & 0 & 6 & -1 & -1 \\ 0 & -1 & -1 & 6 & -1 \\ 0 & 0 & -1 & -1 & 6 \end{pmatrix} \quad ,$$

$$K = \begin{pmatrix} 2 & -2 & 2 & 2 & -2 \\ 2 & 2 & 2 & 2 & 2 \\ 1+\sqrt{13} & 1-\sqrt{5} & 0 & 1-\sqrt{13} & 1+\sqrt{5} \\ 1+\sqrt{13} & -1+\sqrt{5} & 0 & 1-\sqrt{13} & -1-\sqrt{5} \\ 4 & 0 & -2 & 4 & 0 \end{pmatrix} \quad ,$$

$$\text{Eigenvalues: } \quad 6.5+0.5\sqrt{13}, \quad 5.5+0.5\sqrt{5}, \quad 6, \quad 6.5-0.5\sqrt{13}, \quad 5.5-0.5\sqrt{5} \quad .$$

It can easily be verified that the first leading principal minor of K^T is zero:

$$\det \begin{pmatrix} 2 & 2 & 1+\sqrt{13} & 1+\sqrt{13} \\ -2 & 2 & 1-\sqrt{5} & -1+\sqrt{5} \\ 2 & 2 & 0 & 0 \\ 2 & 2 & 1-\sqrt{13} & 1-\sqrt{13} \end{pmatrix} = 0 \quad .$$

The eigenvectors in K are not normal, since we are anyway only interested in whether the principal minor becomes zero or not.
The first two and the last two graphs of figure 6.1 are pairwise isomorphic. As already pointed out in section 4.7 such graphs have the same spectrum, but their node labelling is permutated. As a consequence the columns of K^T are permutated, too, and the zero principal minors become nonzero here. Different initializations do also influence the interchange effect: If Q is a unitary initalization matrix, the QR-algorithm works on the matrix $A_0 = Q^T A_I Q$ instead of $A_0 = A_I$. The effect is not influenced by the use of different upper eigenvalue bounds. They do not change the eigenvectors of L_ρ.
An interesting observation is that if the iteration is not stopped, the interchanged eigenvalues converge after a while to the eigenvalues in their natural order. In figure 6.2 the effect is illustrated. Shown are the Rayleigh-Ritz coefficients of the second graph of figure 6.1 after 85 and 130 iterations. The correct termination lies between 30 and 60. The red eigenvalues are now completely the second-smallest values. The blue eigenvalues are interchanging with the green ones. The larger the distance between disordered eigenvalues, the earlier they are re-ordered. After the re-ordering there is no change in the eigenvalues any more. We argue again from the power iteration viewpoint. Every vector is a linear combination of orthonormal eigenvectors of the iteration matrix. Even if the iteration vectors have converged to eigenvectors with an adequate precision, there remain deviations

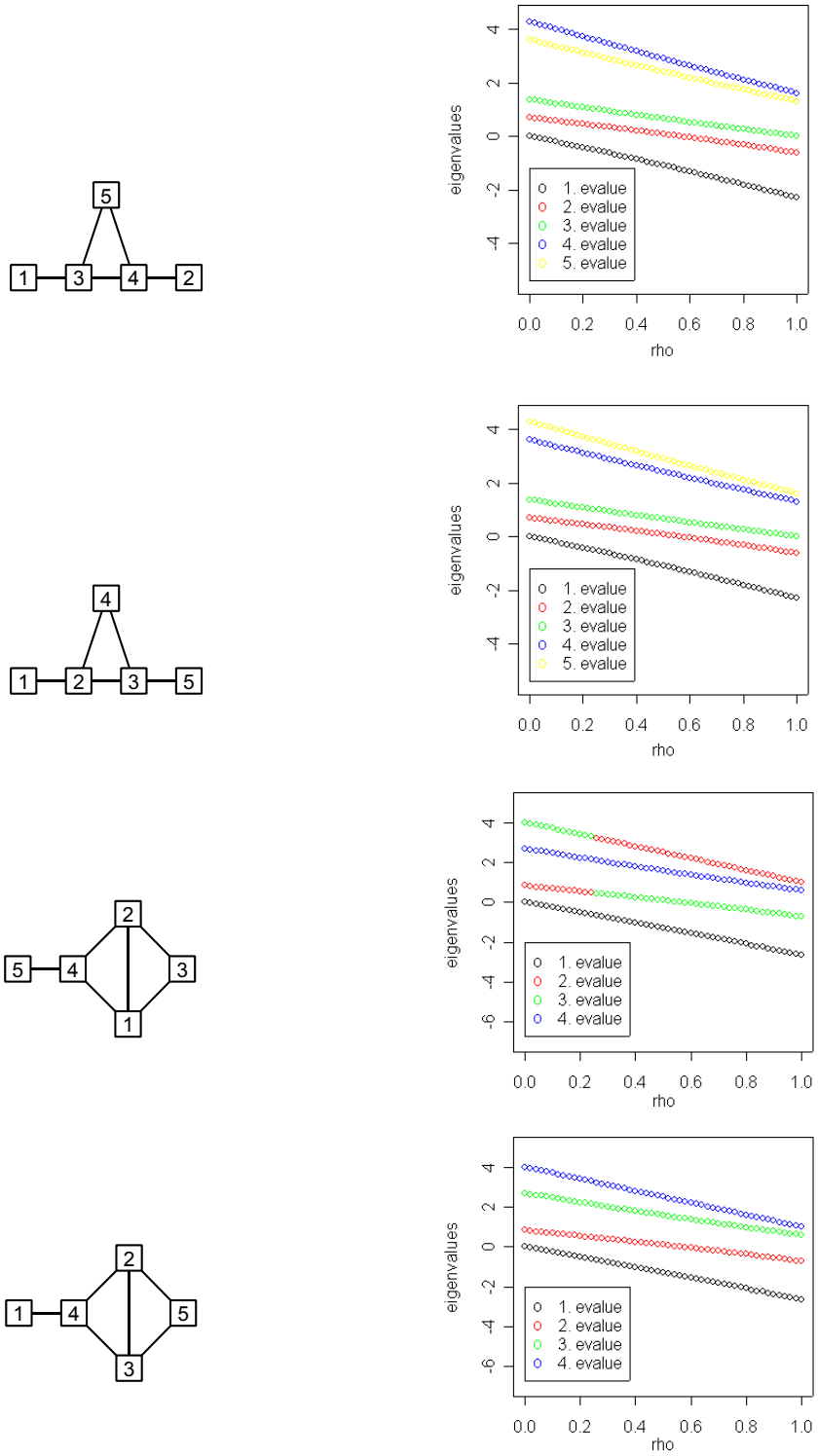


Figure 6.1: The eigenvalue interchange effect

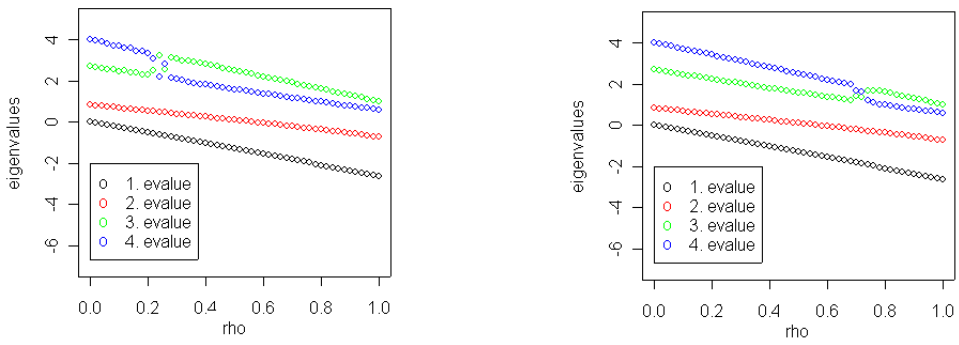


Figure 6.2: Eigenvalue re-ordering. On the left after 85 iterations, on the right after 130 iterations.

in the direction of all other eigenvectors, since calculations on a computer are not exact. Suppose the iteration vector x should converge to the eigenvector a. But because of the interchange effect x has converged to eigenvector b. Then x's deviation in the direction of a will grow exponential in the number of steps, because a's eigenvalue is larger than b's. The vector x will finally converge to a. If another vector has converged to a before, it is orthogonalized to x every step, and will go into another direction. The same may also be forced by the initialization, independet from principal minors. If the start vector for x is very close to b and the deviation to a is very small, then b is dominant first. It is possible, that the algorithm stops before x has converged to a. Figure 7.2 shows an example from dynamic graph drawing. In a number of steps a graph is changed and each time a layout is computed. The eigenvectors of the last step are the start vectors for the next step. In some steps the eigenvalue functions intersect. But for a while the iteration vectors converge to the eigenvectors closest to the initialization. They are not provided in the eigenvalue order. When the distance between the disordered eigenvalues becomes large enough, they are re-ordered.

The eigenvalue interchange effect occurs, but not very frequently and then in most cases between close eigenvalues. The layouts differ, but the energy functions are still minimized - especially for larger graphs - and the layout's quality becomes not too bad. One possibility to counter the interchange effect, is to stop the algorithm a certain amount of iterations after the residuum is small enough. Another is computing some more eigenvectors as needed for the layout and picking the right ones. Both is expensive. A better method is varying the initialization. We had good results with randomly chosen start vectors. We may assume, that random vectors are linear independent and the orthogonalization does not break down. But sometimes also a random initilization is not very advantageous, e.g.

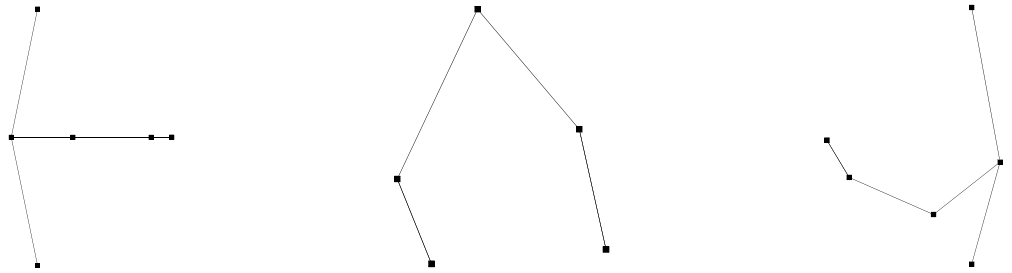


Figure 6.3: Relaxed Laplace layouts of the graph from figure 4.1 for $\rho = \frac{1}{2}\sqrt{2}$.

if speedup techniques should be used. Then there is another common initialization trick: If there are p eigenvalues to compute, we start a one-dimensional iteration with some initialization $q_0^{(1)}$. After we have the first eigenvector $q^{(1)}$, we start a two-dimensional iteration, with start vector $q^{(1)}$ and another vector $q_0^{(2)}$. If we have k eigenvectors, we start $(k+1)$-dimensional iteration, unless $k = p$. To save time, we omit the expensive matrix multiplications with the already converged eigenvectors $q^{(i)}$, $1 \leq i \leq k-1$. In most cases the costs are even lower than in the original orthogonal iteration, because the algorithm stops for every single iterate independently. The method is formalized as algorithm 3 and can be used in algorithm 2 as alternative to procedure ITERATE. The advantage of this procedure is, that the iteration vectors are projected in subspaces, which are orthogonal to already converged eigenvectors. Deviations in these directions are minimized. Since the first eigenvector does not interchange, the following iteration vectors are as in a induction all more likely to be in the right order. The procedure is more robust against interchange effects, but not completely free of them, see figure 7.2. And there is no re-ordering possible.

Another kind of anormaly may occur, if graphs are laid out, which have multiple eigenvalues. The layout then differs depending on the initialization. There is a multi-dimensional subspace of possible eigenvectors. The algorithm will converge to those vectors, that are the closest to the start vectors. Let us again consider the graph from figure 4.1. We showed that for $\rho = \frac{1}{2}\sqrt{2}$ the graph has a double eigenvalue. In figure 6.3 some corresponding layouts are represented. The left and the middle layout have in y-dimension the in section 4.4 computed eigenvectors as axis vectors, the right one a linear combination of them. For the left and the middle layout we initialized the algorithm with unit vectors, for the right with random vectors.
A spectral layout of a graph G needs not to be unique, if the matrix has multiple eigenvalues. We show now, that there needs not to be an unique graph belonging to a certain

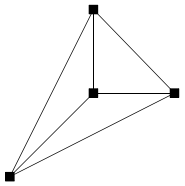


Figure 6.4: Equal Laplace layout of two different graphs.

layout, too. Given are the axis vectors of a 2-dimensional Laplace layout:

$$x = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ -1 \\ 0 \end{pmatrix}, \quad y = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ -1 \\ 0 \end{pmatrix} \quad .$$

Since the layout is a Laplace layout, $\mathbf{1}$ is another eigenvector. We found two distinct graphs, whose eigenvectors corresponding to the smallest eigenvalues of the Laplace matrix equal $\mathbf{1}$, x and y:

$$L(G) = \begin{pmatrix} 5 & -1 & -1 & -1 & -1 & -1 \\ -1 & 5 & -1 & -1 & -1 & -1 \\ -1 & -1 & 5 & -1 & -1 & -1 \\ -1 & -1 & -1 & 5 & -1 & -1 \\ -1 & -1 & -1 & -1 & 5 & -1 \\ -1 & -1 & -1 & -1 & -1 & 5 \end{pmatrix}, \quad L(G') = \begin{pmatrix} 11 & -1 & -1 & -4 & -1 & -4 \\ -1 & 11 & -1 & -4 & -1 & -4 \\ -1 & -1 & 11 & -4 & -1 & -4 \\ -4 & -4 & -4 & 20 & -4 & -4 \\ -1 & -1 & -1 & -4 & 11 & -4 \\ -4 & -4 & -4 & -4 & -4 & 20 \end{pmatrix} \quad .$$

The corresponding eigenvalues of G are $\{0, 6, 6\}$ and of G' are $\{0, 12, 12\}$. The spectrum of G is $\{0, 6, 6, 6, 6, 6\}$ and of G' is $\{0, 12, 12, 12, 24, 24\}$. Both are *complete graphs*, every node is adjacent to any other. In G all edges are weighted equal, while in G' the edges of two nodes are more emphasized than the others. Even a 3-dimensional layout equal for both can be constructed. Figure 6.4 shows the (not very representing) Laplace layout of G and G'. Also for the other spectral layout in the next sections such examples can be constructed.
We finally remark that depending on whether the algorithm provides vector x or vector $-x$, the layout may be reflected by the corresponding axes.

Algorithm 3 Alternative Procedure ITERATE

With the same notation as in algorithm 2 and additionally $x_k \in \mathbb{R}^n$:

procedure ITERATE(A, Q_0, p)
  for $j = 1, ..., p$ **do**
    $x_0 = q_0^{(j)}$
    repeat
      $x_k = Ax_{k-1}$
      $r(A, x_{k-1}) = || x_k - \left((x_{k-1})^T x_k\right) x_{k-1}||^2$
      $x_k = x_k - \sum_{i=1}^{j-1} \left((q^{(i)})^T q^{(j)}\right) q^{(i)}$
      $x_k = \frac{x_k}{|| x_k ||}$
    until $\max_{1 \leq j \leq p} r(A, x_{k-1}) \leq \delta$ or $k > k_{\max}$
    $q^{(j)} = x_k$
  end for
  return $Q = \left(q^{(1)}, ..., q^{(p)}\right)$
end procedure

Chapter 7

Dynamic Graph Drawing Using Spectral Layouts

In some applications graphs must be drawn, that change over time. This is called *dynamic graph drawing*. Think of the internet backbone, where we have traffic (edges) between servers (nodes). Both the amount of traffic and the number of server vary. Using spectral methods for dynamic graph drawing is a new approach. To illustrate the possibilities we made an 3D-animation called *Spectral Dance*, in which a grid gets folded, broken apart, rotated and finally becomes the original grid again.
The animation was displayed on the facade of the Museum of the Future during a conference[1] and exhibition[2] on the Language of Networks, which was part of the Ars Electronics Festival 2004 in Linz. It also won in the category free-style the graph drawing contest of the 12th International Symposium on Graph Drawing 2004 in New York [GD]. The animation and the Language of Networks conference trailer, of which the animation is a part, are on the enclosed CD. Figure 7.1 shows two screenshots of the animation and a photo of the museum during the Language of Networks conference.

To introduce the basic concepts of dynamic graph drawing, we summarize [KW, pp. 228-230]: The straightforward solution to draw graphs that change over time is to compute every step independently with an static algorithm. But there arise two drawbacks: On the one hand, the graph is often only slightly changed in one step. So the information of the old graph should be reused to decrease computation costs. On the other hand, even slightly changes may produce completely different looking layouts. The viewer is familiar with the old drawing, he has build a so-called *mental map*. So the new drawing should be very similar to the old, to help the viewer recognizing the graph's structure and its

[1] `http://www.aec.at/en/festival/programm/list_conferences_2004.asp?iProjectID=12570`
[2] `http://www.aec.at/en/festival/programm/project_2004.asp?iProjectID=12654`

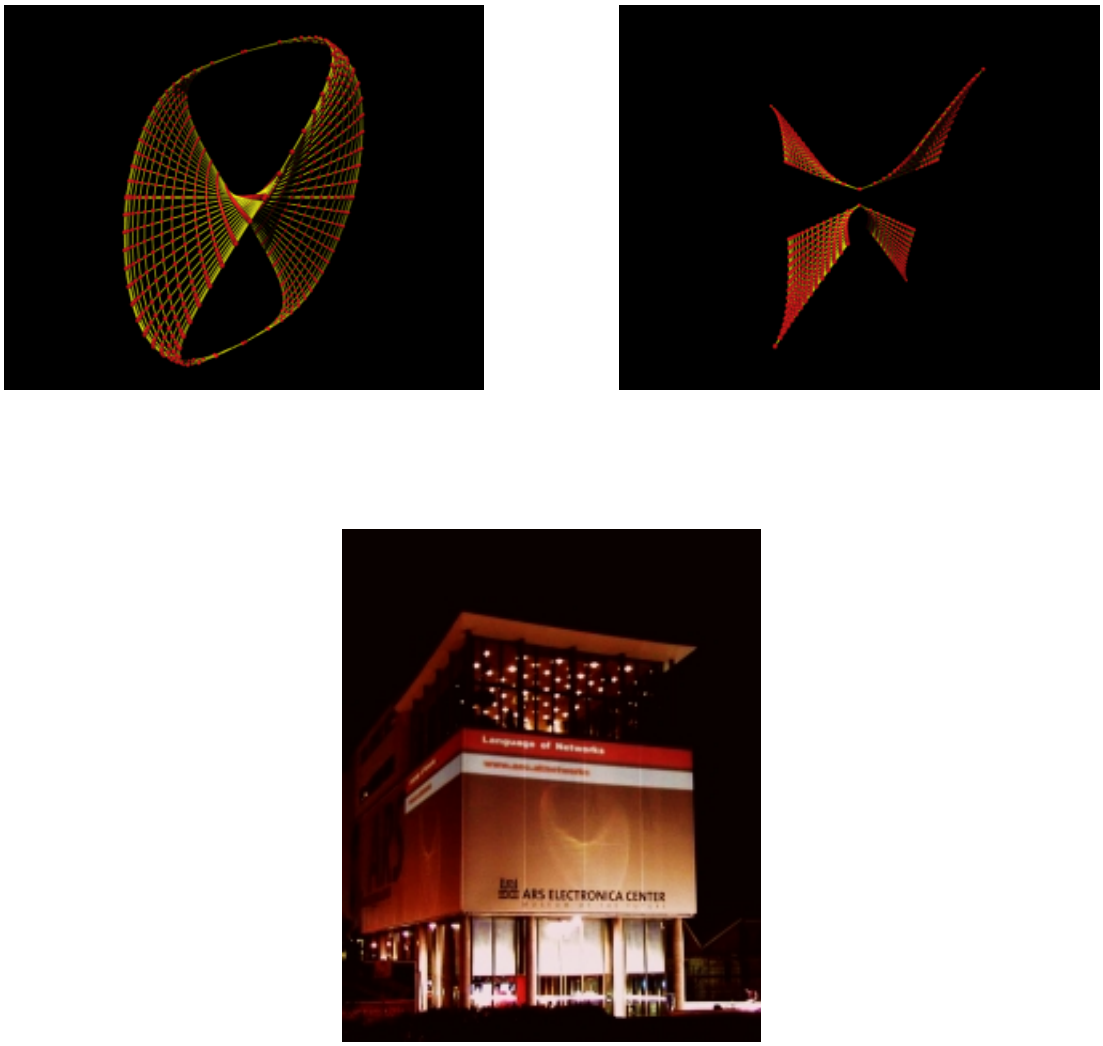
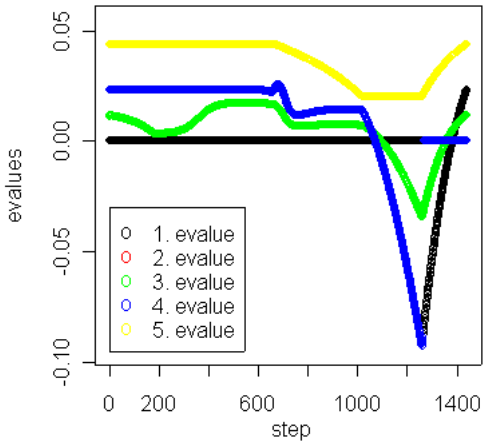


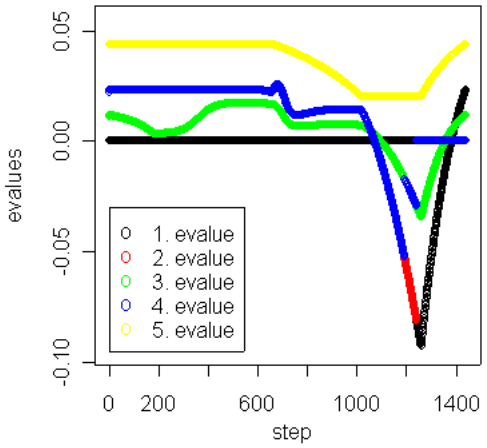
Figure 7.1: The animation Spectral Dance. Screenshots and a photo of the facade of the Museum of the Future during the conference Language of Networks 2004 in Linz.

changes. A common way to maintain the mental map is first to compute a layout, that compromises between the original layout goals and minimizing the changes. And in a second step the changes are animated.

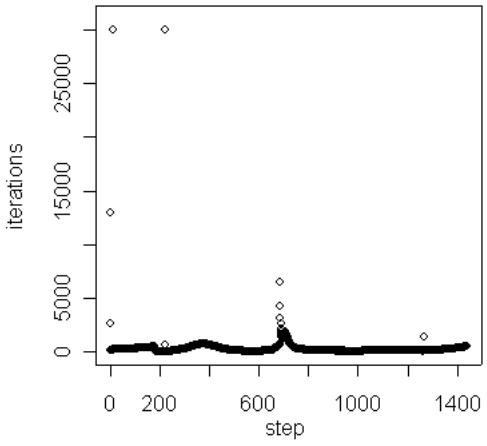
We now describe how we adapted spectral layouts to the concepts of dynamic graph drawing for the Spectral Dance. We stated in condensed form a recipe for the general case as algorithm 4. If a graph is changed, also its related matrices are changed and hence the spectrum of the graph. We will use the fact that continuous perturbations in a graph related matrix cause continuous eigenvalues and in practice continuous eigenvectors, except for some special cases (see section 3.3). This property ensures that slightly changes in the graph cause very similar looking layouts and the viewer's mental map can be preserved. The eigenvectors of the Laplace matrix are the best choice for drawing highly structured graphs like grids, so we took them for the Spectral Dance. Alternatively the generalized or the relaxed Laplace layout may be used. To use the generalized layout the iteration vectors must be multiplicated with $D^{\frac{1}{2}}$ before applying the procedure ITERATE. After the procedure they must be multiplicated $D^{-\frac{1}{2}}$. We started with a 30×30 grid and manipulated every step the weights of the graph and its Laplace matrix L. Then we computed a Laplace layout using algorithm 2. The results are stated in figure 7.2. We used some negative weights and therefore had for a period negative eigenvalues. We chose the dimension larger than three because of possible interchange effects. To save computation time we set as start vectors the eigenvectors of the last step. They were a good approximation and the number of iterations decreased a lot. For the first step we needed about 30000 iterations. This would also have been the magnitude of iterations for the other steps. But so we had as mean about 300 iterations for eigenvectors with the same precision. Outliers coincided with eigenvector interchange effects and the local maximum at step 660 with an increased change in the data. If the change in the data is moderate, then acceleration methods need not to be applied in general. After the eigenvectors were computed, the right ones for the layout had to be picked. For the first layout we started, according to the Laplace layout, with the eigenvectors corresponding to the smallest non-trivial eigenvalues. From then on in every new step and every layout dimension the eigenvector that is closest to the old eigenvector must be chosen to obtain continuous drawings. This is necessary, since caused by interchange effects the order in which the algorithm provides the eigenvectors may be altered. Zero principal minors are possible and eigenvalue functions could intersect. After enough transactions in the graph this seems to be inevitable. There may even be eigenvectors whose eigenvalues become smaller than the eigenvalues of axis vectors. But we have to accept this as a compromise between the spectral layout goals and minimizing the changes in the layout. As an effect of the initialization the eigenvectors of multiple eigenvalues are continuous, if the dimension of the eigenspace does not change. If there are intersections in the eigenvalue functions, eigenvectors may be discontinuous (see theorem 3.13). If the matrix entries are changed linearly, this is a very rare event. A linear interpolation would be a simple



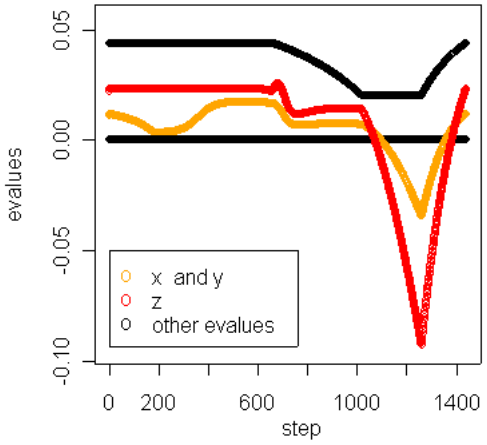
original ITERATE



modified ITERATE



number of iterations



chosen axis vectors

Figure 7.2: Eigenvalues of the Spectral Dance computed using algorithm 2. Top: Comparison of both ITERATE procedures. Bottom: The number of iterations and the chosen axis vectors.

workaround. There were indeed some intersections in our computations for the Spectral Dance, but they did not cause discontinuities. With the described methods our layouts were continuous.
We tested both ITERATE version. Running times were about the same. The modified version tends more to re-order eigenvalues after intersections. This confirms what we already pointed out in section 6.2. The original version is in this scope probably better. If the eigenvectors are picked manually, then there is less re-ordering and so less interchanges in the original version. Let p be the number of computed eigenvectors. If the eigenvalues of an axis vector become larger than the first p eigenvalues, then the original vector may for a longer period still provide the axis vector than the modified version.
In the Spectral Dance we implicitly added and deleted edges by changing the weight. The number of nodes remained constant and the graph connected. But in general it is possible, that nodes are added or deleted or the graph is not connected any more. Therefore we need the following theorem:

Theorem 7.1
Given is a graph $G = (V, E, \omega)$. The spectrum of a graph related matrix from chapter 4 of G is the disjunct union of the spectra of the graph's connected components.

Proof:
Suppose the graph G consists of two connected components. Then G is isomorphic to a graph with the following adjacency matrix $A \in \mathbb{R}^{n \times n}$:

$$A = \begin{pmatrix} B & 0 \\ 0 & C \end{pmatrix} \quad ,$$

with $B \in \mathbb{R}^{l \times l}$, $C \in \mathbb{R}^{m \times m}$ and m, l nonzero, $l + m = n$. B and C are the adjacency matrices of the two connected components. With some elementar linear algebra follows, that the charactersistic polynomial of A is the product of the characteristic polynomials of B and C. So the set of eigenvalues of A is the disjunct union of the sets of eigenvalues of B and C. This follows analogue for more connected components than two and for the other graph related matrices. Additionally, if $b \in \mathbb{R}^l$ and $c \in \mathbb{R}^m$ are eigenvectors of B and C, then $(b, 0, ..., 0)^T \in \mathbb{R}^n$ and $(0, ..., 0, c)^T \in \mathbb{R}^n$ are eigenvectors of A. □

So if the graph is resolved into several connected components, they are laid out independly. If a node should be added, then in the iteration matrix a zero row and a zero column and in the iteration vectors one zero entry must be added. In a few steps the

edge weights of the new node can be increased up to the desired amount and the layouts remain continuous. Deleting nodes works similar: The weights must be decreased and finally the corresponding zero row and zero column in the matrix and the zero entries in the vectors must be removed.
The number of intermediate steps in algorithm 4 should depend on the change in the graph. In practice it is more economic to compute less steps and to animate the changes with interpolation methods. Our intention was to explore the possibilities of spectral layouts for dynamic graph drawing. So we animated the Spectral Dance just by computing many slightly changed spectral layouts - with an encouraging result.

Algorithm 4 Dynamic Spectral Layout

Given is a graph $G(t) = \big(V(t), E(t), \omega(t)\big)$, a graph related matrix $A(t) = \big(a_{ij}(t)\big) \in \mathbb{R}^{|V(t)|\times|V(t)|}$ with p eigenvectors $Q(t) \in \mathbb{R}^{|V(t)|\times p}$ depending on a factor $t \in \mathbb{R}$ and the number of intermediate steps $s \geq 1$.

procedure COMPUTE STEP$\big(G(t_0),\ Q(t_0),\ G(t_1),\ s\big)$
  If nodes are added between t_0 and t_1, add zero rows and columns to $A(t_0)$ and zero rows to $Q(t_0)$.
  for $(r = 1,\ r \leq s,\ r{+}{+})$ **do**
    $t_{\text{old}} = t_0 + \frac{r-1}{s}(t_1 - t_0)$
    $t_{\text{new}} = t_0 + \frac{r}{s}(t_1 - t_0)$
    $a_{ij}(t_{\text{new}}) = a_{ij}(t_0) + \frac{r}{s}(a_{ij}(t_1) - a_{ij}(t_0))$
    $Q(t_{\text{new}})$ = ITERATE ($A(t_{\text{new}})$,$Q(t_{\text{old}})$,p)
    Pick the vectors from $Q(t_{\text{new}})$, that are closest to the last axis vectors as new axis vectors. If the eigenvectors are discontinuous, interpolate between continuous intervals.
  end for
  If nodes are deleted between t_0 and t_1, remove zero rows and columns from $A(t_1)$ and zero rows from $Q(t_1)$.
  return $Q(t_1)$
end procedure

Chapter 8

Conclusion

The intention of this thesis is to show that spectral layouts are a valuable implementation of the force-directed graph drawing idea. Although the origins go back to a work of Tutte in the 1970's, there was no further development until the last few years.
We presented two modifications of the classical Laplace layout. The class of graphs, which can reasonable be laid out, is extended by them from some well-structured graphs to the most real world graphs. Especially the relaxed Laplace layout was part of our studies, since there is no sufficient characterization in literature up to now. A main question was necessarily, how the varying degree matrix influences the layout. We computed the energy functions and stated an heuristic for the relaxation factor. Further we found an upper bound for the similarity between Laplace and relaxed Laplace layouts and improved the eigenvalue bounds of L_ρ. We also characterized the generalized Laplace layout and showed the similarity to the relaxed layout.
We used the orthogonal iteration to compute eigenvectors, as proposed from the most authors. During the work on this thesis we observed remarkable convergence anormalies, which did not fit in the usual power iteration framework. After some investigation we found the reason in Wilkinsons proof of the QR-iteration from the 1960's. With this theory we have an exact convergence proposition. The particular procedures of our algorithm have a flexible design, so that acceleration methods can be applied and dynamic graphs can be drawn. We introduced the concept of the normalized Laplace matrix to be able to compute also generalized eigenvectors with the same iteration.
Finally we applied the concepts of dynamic graph drawing to spectral layouts and had very encouraging results. The layouts were continuous and the computation times very fast.

Some work is still to do. We stated a well-working heuristic for the relaxation factor ρ. But ρ could be optimized to further, more general constraints. A related unsolved

problem is the question, whether relaxed and generalized Laplace layout are equal for some ρ. The first steps are already done. The similarity is shown and a necessary condition is found, which is easy to test. We hope that some time also the second steps are done.

Appendix A

Content of the Enclosed CD

On the enclosed CD there are two video files, the original version of the Spectral Dance and the Language of Networks trailer, in which the animation is shown in the background. It was our intention to keep all test results of this work reproducable. Therefore the graphs from figure 5.3 and 5.4 are included in different file formats. All other test graphs are small and can easily be generated by hand. We also added our implementation of the relaxed Laplace and the generalized Laplace layout for the yEd graph editor. It requires minimal Java™ version 1.4, yEd version 2.2 and yFiles version 2.2. In both layouters it can be chosen between a random or unit vector initialization, between the two different variants of the ITERATE procedure and whether self-loops should be ignored. The number of eigenvectors and the maximum number of steps can be set. For the relaxed layout can additionally the upper Brauer eigenvalue bound be used, a value for ρ be set or the heuristic ρ be taken. There is also another implementation of the relaxed layout, which computes the eigenvalue functions for $\rho \in [0, 1]$, to detect, whether there are intersections.

There are several graph generators provided by yEd. To have enough test instances we additionally implemented three generators, which are also included on the CD. The simplest is the $G(n, p)$ model. The number of nodes n and the probability p is given. Each possible edge is inserted with probability p independent from all other edges.
The second generates *scale-free* graphs, i.e. graphs whose degree distribution follows a power law. The probability for degree d_i is $\frac{1}{d_i^\gamma}$ for a $\gamma > 0$ in such a graph. We used the method described in [BR]: The number of nodes n and the outdegree d are given. Every step we add one node and d edges from that node to other nodes, which are already there. The edges are added randomly, but the probabilities for the nodes corresponds with their degree. Edges prefer to connect to nodes with high degrees. This is called *preferential attachment*. We start with one node and stop when all nodes are inserted. Self-loops and multiple edges are allowed and emerge necessarily. To have reasonable results we should

set $d << n$.
The third generator creates *small world* graphs: Such graphs have low deviations in the degree and are only locally dense, but also the shortest paths between all nodes are relative short. We use the idea of [WS]: Given is the number of nodes n, the neighborhood parameter $r < \frac{n-1}{2}$ and a probability p. First a (n,1,r)-torus of nodes is built. Then with probability p each edge is disconnected from one of its nodes and reconnected with a random node. For well-structured graphs as in figure 5.3 the probability p should be set very small.

Bibliography

[AM] W. Anderson and T. Morley, Eigenvalues of the Laplacian of a graph, in *Lin. Multilin. Algebra 18*, pp. 141-145, Gordon & Breach, 1985.

[BETT] G. Di Battista, P. Eades, R. Tamassia and I. Tollis, *Graph Drawing: Algorithms for the Visualization of Graphs*, Prentice-Hall, 1999.

[BP] A. Berman and R. Plemmons, Nonnegative matrices in the mathematical sciences, Academic Press, New York, 1979.

[BR] B. Bollobas and O. Riordan, The diameter of a scale-free random graph, Combinatorica 24, pp. 5-34, 2004.

[BS] U. Brandes and S. Cornelsen, Visual Rank of Link Structures, Journal of Graph Algorithms and Applications, vol. 7, no. 2, pp. 181-201, 2003.

[BW] U. Brandes and T. Willhalm, Visualizing Bibliographic Networks with a Reshaped Landscape Metaphor, *Proc. 4th Joint Eurographics - IEEE TCVG Symp. Visualization (Vis-Sym'02)*, pp. 159-164, ACM Press, 2002.

[Ch] F. Chung, Spectral Graph Theory, Number 52 in Conference Board of the Mathematical Sciences, American Mathematical Society, 1994.

[Fie] M. Fiedler, Special matrices and their applications in numerical mathematics, Martinus Nijhoff Publishers and SNTL, 1986.

[Fis] G. Fischer, Lineare Algebra, Vieweg, 12. Auflage, 2000.

[GD] Franz J. Brandenburg, Christian A. Duncan, Emden Gansner, and Stephen G. Kobourov, *Graph-Drawing Contest Report Proc. 12th Intl. Symp. Graph Drawing (GD'04)*, Lecture Notes in Computer Science 3383, Springer-Verlag, 2005, To appear.

[GR] C. Godsil, G. Royle, Algebraic graph theory, Springer, 2001.

[GvL] G. Golub, C. van Loan, Matrix Computations, John Hopkins University Press, 3. edition, 1996.

[HoJo] R. Horn and C. Johnson, Matrix Analysis, Cambridge University Press, 1996.

[KCH] Y. Koren, L. Carmel and D. Harel, Drawing Huge Graphs by Algebraic Multigrid Optimization, Multiscale Modeling and Simulation, Volume 1, no. 4, pp. 645-673, SIAM, 2003.

[Ko] Y. Koren, Drawing Graphs by Eigenvectors: Theory and Practice, Computers and Mathematics with Applications, Elsevier, to appear.

[KW] M. Kaufmann and D. Wagner (Eds.), Drawing Graphs: Methods and Models, LNCS 2025, Springer Verlag, 2001.

[Mo91] B. Mohar, The laplacian spectrum of graphs, in *Graph Theory, Combinatorics, and Applications*, editors Y. Alavi, G. Chartrand, O. R. Oellermann, A. J. Schwenk, Vol. 2, pp. 871-898, Wiley, 1991.

[Mo97] B. Mohar, Some applications of Laplace eigenvalues of graphs, in *Graph Symmetry: Algebraic Methods and Applications*, editors G. Hahn and G. Sabidussi, pp. 225-275, NATO ASI Ser. C 497, Kluwer, 1997.

[Pa] B. Parlett, The Symmetric Eigenvalue Problem, Prentice-Hall, 1980.

[Va] R. Varga, Matrix Iterative Analysis, Springer, Second Edition, 2000.

[Wi] J. Wilkinson, The Algebraic Eigenvalue Problem, Oxford University Press, 1965.

[WS] D. Watts and S. Strogatz, Collective dynamics of small-world networks, Nature 393, pp. 440-442, 1998.